

Міністерство освіти і науки, молоді та спорту України  
Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії

«До захисту допущено»  
Завідувач кафедри  
комп'ютерної інженерії  
к.т.н., доц. О.М. Березький

“ \_\_\_\_\_ ” \_\_\_\_\_ 201\_\_ р.

## **ДИПЛОМНИЙ ПРОЕКТ**

освітньо-кваліфікаційного рівня "Спеціаліст"  
зі спеціальності 7.05010201 "Комп'ютерні системи та мережі"  
на тему:

### **АРИФМЕТИЧНИЙ ПРИСТРІЙ RISC-ПРОЦЕСОРА В СЕРЕДОВИЩІ XILINX ISE DESIGN SUITE**

Студент групи КСМс-51 \_\_\_\_\_ Морозова А.В.  
(підпис)

Керівник:  
к.т.н., доцент \_\_\_\_\_ Ляпандра А.С.  
(підпис)

Нормоконтроль:  
к.т.н., доцент \_\_\_\_\_ Васильків Н.М.  
(підпис)

Консультант  
з охорони праці:  
доцент \_\_\_\_\_ Сапожник Г.В.  
(підпис)

Міністерство освіти і науки, молоді та спорту України  
Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії  
Спеціальність 7.05010201 “Комп'ютерні системи та мережі”

“Затверджую”  
Завідувач кафедри  
комп'ютерної інженерії  
к.т.н., доц. О.М. Березький

\_\_\_\_\_ 201\_\_ р.  
“ \_\_\_ ” \_\_\_\_\_

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТА

Морозова Антоніна Вікторівна

(прізвище, ім'я по-батькові)

- 1. Тема проекту** “Арифметичний пристрій RISC-процесора в середовищі Xilinx ISE Design Suite” затверджена наказом університету від “ \_\_\_ ” \_\_\_\_\_ 201\_\_ р. №\_\_
- 2. Термін здачі студентом закінченого проекту** « \_\_\_ » \_\_\_\_\_ 2012 р.
- 3. Вихідні дані до проекту:** Технічне завдання.
- 4. Перелік задач, які мають бути вирішені:**
  - провести аналіз існуючих рішень для програмно-апаратного шифрування даних;
  - визначити оптимальну структурну організацію програмно-апаратної системи для шифрування даних;
  - вибрати метод шифрування даних;
  - розробити алгоритм шифрування даних;
  - розробити архітектуру апаратної частини системи;
  - виконати програмну реалізацію алгоритму шифрування даних;
  - виконати тестування програмної частини системи;
  - оцінити швидкодію програмно-апаратної системи шифрування даних;
  - виконати порівняльний аналіз розробленої системи з існуючими програмно-апаратними засобами для шифрування даних.

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

- Архітектура системи. Схема структурна
- Взаємодія програмних модулів. Схема структурна
- Шифрування даних. Схема алгоритму
- Дешифрування даних. Схема алгоритму
- Модуль шифрування. Схема електрична функціональна
- Блок кодування. Схема електрична принципова

**6. Консультанти по проекту (із зазначенням розділів):**

Розділ	Консультант	Завдання видав	Завдання прийняв
Охорона праці	Сапожник Г.В.		

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва розділів дипломного проекту	Термін виконання	Позначки керівника про виконання завдань
1.	<b>Ошибка! Источник ссылки не найден.</b>	15.09.2011 – 5.11.2011	
2.	<b>Ошибка! Источник ссылки не найден.</b>	6.11.2011 – 31.12.2011	
3.	<b>Ошибка! Источник ссылки не найден.</b>	1.01.2012 – 14.03.2012	
4.	Охорона праці	15.03.2012 – 23.03.2012	

**Завдання прийняв до виконання** \_\_\_\_\_  
(підпис)

**Керівник дипломного проекту** \_\_\_\_\_  
(підпис)

## АНОТАЦІЯ

Робота виконана на 106 сторінках, містить 23 рисунки, 14 таблиць, 8 додатків, з них 4 графічного матеріалу.

Метою дипломного проекту є створення комплексного підходу до практичної реалізації арифметичного пристрою RISC-процесора.

Актуальність розроблення такого пристрою не викликає сумніву. Арифметичний пристрій RISC-процесора в середовищі Xilinx ISE Design Suite призначений для використання у блоці арифметико-логічних операцій реконфігурованого мікропроцесорного ядра. Він містить в собі декілька підсистем, розробка яких і проведена в даному дипломному проекті.

Даний блок розроблений з дотриманням вимог, які ставляться до таких розробок, обґрунтовано вибір частин усіх блоків, а тому результат роботи може бути використаний у навчальному процесі.

## ANNOTATION

Work is executed on 106 pages, contains 23 figures, 14 tables, 8 additions, from them 4 graphic material.

The aim of the diploma project is to create a comprehensive approach to the practical implementation of arithmetic device RISC-processor.

Urgency of developing such a device is beyond doubt. Arithmetic device RISC-processor environment in Xilinx ISE Design Suite is intended for use in arithmetic logic unit operations reconfigurable microprocessor core. It contains several subsystems, development and held that in this thesis project.

This unit is designed in compliance with requirements that relate to these developments, the choice parts of all blocks, so the result of work can be used in the learning process.

# ТЕХНІЧНЕ ЗАВДАННЯ

## 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Арифметичний пристрій RISC-процесора в середовищі Xilinx ISE Design Suite.

1.2 Область застосування – комп'ютерні системи з реконфігурованою архітектурою

## 2. ОСНОВА ДЛЯ РОЗРОБКИ

Основою для розробки є завдання на дипломний проект, затверджене кафедрою комп'ютерної інженерії факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету.

## 3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою дипломного проекту є розробка арифметичного пристрою RISC-процесора

## 4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами даної розробки є матеріали навчальної та реферативної наукової літератури, технічна документація, науково-дослідні роботи, журнали.

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1 Вимоги до апаратних засобів:

#### 5.1.1 Функціональні вимоги

5.1.1.1 Арифметичний пристрій RISC-процесора в середовищі Xilinx ISE Design Suite призначений для застосування у мікропроцесорних ядрах побудованих на реконфігурованих елементах.

5.1.1.2 Умови роботи досліджуваного арифметичного пристрою RISC-процесора – штатні, згідно з вимогами до ПЛІС.

#### 5.1.2 Вимоги до параметрів

5.1.2.1 Сигнальний стандарт логічних рівнів введення-виведення – LVCMOS33.

5.1.2.2 Максимальна кількість логічних комірок арифметичного пристрою RISC-процесора – 2160.

### 5.1.3 Вимоги до елементної бази

5.1.3.1 Програмована логічна інтегральна схема типу XC3S100E CP132.

### 5.1.4 Вимоги до надійності

5.1.4.1 Середня кількість реконфігурування ПЛІС – 10000 разів.

### 5.1.5 Вимоги безпеки

5.1.5.1 Технічне забезпечення системи повинно відповідати вимогам електробезпеки (ГОСТ 25.861-85).

### 5.1.6 Умови експлуатації

5.1.6.1 Мікроклімат в приміщеннях повинен відповідати нормам виробничого мікроклімату для обчислювальних центрів (ГОСТ 12.1.005-88)

5.1.6.2 Для нормальної роботи системи необхідно підтримувати (по ГОСТ 23.865-85):

- температуру повітря в межах від +18°C до +28°C;
- відносну вологість повітря при +25°C в межах від 40% до 80%;
- атмосферний тиск  $760 \pm 25$  мм.рт.ст..

## **5.2 Вимоги до програмної системи**

### 5.2.1 Функціональні вимоги

5.2.1.1 Вхідна інформація отримується шляхом:

- подачі цифрових сигналів на вхід з генератора випадкових сигналів.

5.2.1.2 Вихідна інформація:

- вихідна інформація виводиться на чотири семисегментні індикатори.

### 5.2.3 Вимоги до програмного забезпечення:

5.2.3.1 Операційна система сімейства Windows;

5.2.3.2 Програма Xilinx ISE Design Suite (версії не нижче 10.1);

5.2.3.3. Обсяг конфігураційного файлу – до 581344 біт.

### 5.2.4 Вимоги до програмної документації

5.2.4.1 Блок – схема алгоритму;

5.2.4.2 Програмний код.

## **7. ВИМОГИ ОХОРОНИ ПРАЦІ**

В розділі “Охорона праці” дипломного проекту повинен бути даний аналіз умов праці в приміщенні.

## **8. ПОРЯДОК КОНТРОЛЮ І ПРИЙОМКИ**

8.1 Представлення дипломного проекту на попередній захист.

8.2 Представлення дипломного проекту на захист.



## ЗМІСТ

Перелік умовних позначень .....	12
Вступ.....	13
1 Арифметично-логічні пристрої ЕОМ.....	14
1.1 Призначення, склад і структура АЛП .....	14
1.2 Класифікація арифметично-логічних пристроїв ЕОМ.....	17
2 Засоби представлення арифметично-логічних пристроїв ЕОМ.....	26
2.1 Узагальнені функції систем переробки інформації .....	30
2.2 Загальна методологія та етапи проектування АЛП .....	34
3 Загальний порядок проектування АЛП на основі функціонально-структурного підходу.....	47
3.1 Формування функціональної структури АЛП .....	47
3.2 Модифікація алгоритмів і структур АЛП.....	63
3.3 Оцінка структур АЛП .....	72
3.4 Дослідження структурної моделі арифметичного пристрою .....	78
4 Охорона праці .....	80
4.1 Аналіз санітарно-гігієнічних умов праці .....	80
4.2 Пожежна безпека.....	92
Висновки .....	94
Список використаних джерел .....	95
Додаток А. Арифметичний пристрій RISC-процесора. Схема структурна .....	97
Додаток Б. Арифметичний пристрій RISC-процесора. Схема функціональна ...	98
Додаток В. Арифметичний пристрій RISC-процесора. Схема електрична принципова.....	99
Додаток Г Арифметичний пристрій RISC-процесора.....	100
Додаток Д. Арифметичний пристрій RISC-процесора. Лістинг. ....	101

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>				
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>					
<i>Розроб.</i>		<b>Ошибка!</b>			<b>Ошибка! Источник ссылки не найден.</b>	<b>Літ.</b>	<b>Арк.</b>	<b>Аркушів</b>	
<i>Перевір.</i>		<b>Ошибка!</b>						9	<b>Ошибка!</b>
<i>Конс.</i>		<b>Ошибка!</b>				<b>THEU.FKIT.Ошибка!</b>			
<i>Н. Контр.</i>		<b>Ошибка!</b>							
<i>Затверд.</i>		<b>Ошибка!</b>							

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CPLD	–	складний програмований логічний пристрій;
FPGA	–	програмована матриця логічних елементів;
HDL	–	мова опису обладнання та апаратного забезпечення;
ISE	–	інтегроване середовище синтезу;
АЛП	–	арифметико-логічний присистрій;
АЦП	–	аналого-цифровий перетворювач;
БМК	–	базовий матричний кристал;
Діла	–	ділянка алгоритму;
ІФО	–	інформаційні об'єкти;
КС	–	комп'ютерна система;
ОЗП	–	оперативний запам'ятовувальний пристрій;
ПК	–	персональний комп'ютер;
ПЛІС	–	програмована логічна інтегральна схема;
ППЗП	–	постійний програмований запам'ятовувальний пристрій;
СОІ	–	система опрацювання інформації;
ТБ	–	типовий блок
ТЗ	–	технічні засоби;
ТУ	–	технічні умови;
УФБ	–	узагальнений функціональний блок;
ФСП	–	функціонально-структурний підхід.

## ВСТУП

Всі основні операції по перетворенню даних в ЕОМ проводяться в операційних блоках, які в більшості випадків називаються арифметично-логічним пристроєм. Набір операцій, виконуваних АЛП універсальних ЕОМ, повинен бути функціонально повним, тобто, забезпечувати реалізацію будь-якого обчислювального алгоритму. І хоча функціональну повноту можна забезпечити дуже вузьким набором операцій [9], кількість різних операцій, що виконуються в АЛП, зазвичай складає від декількох десятків до декількох сотень [22]. Це забезпечує скорочення довжини програм і підвищення швидкодії ЕОМ в цілому.

Як правило, в будь-якому АЛП передбачена можливість виконання чотирьох основних арифметичних операцій, декількох логічних операцій, а також зсувів. Набір операцій АЛП є однією з основних його характеристик.

Оскільки АЛП є закінченим у функціональному відношенні пристроєм, то на нього розповсюджуються загальні закономірності технічних систем. Тому у складі АЛП в загальному випадку можна виділити чотири групи вузлів, відповідних основним системним процесам: зберігання, передачі, перетворення, управління.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1 АРИФМЕТИЧНО-ЛОГІЧНІ ПРИСТРОЇ ЕОМ

У цьому розділі розглядаються загальні питання, пов'язані з призначенням, організацією та основними характеристиками АЛП, а також приводиться їх класифікація. Крім того, даються відомості про засоби опису АЛП і процесів їх функціонування.

### 1.1 Призначення, склад і структура АЛП

До вузлів зберігання в АЛП відносяться:

- реєстри, що забезпечують зберігання операндів, проміжних і остаточних результатів;
- тригери, що дають можливість зберігати різні ознаки результатів або які-небудь допоміжні біти.

В деяких випадках реєстри АЛП утворюють блок реєстрової пам'яті, а тригери (звані також прапорцями) об'єднуються в реєстр стану.

До вузлів передачі, наявних в АЛП, відносяться:

- шини, що сполучають окремі блоки АЛП;
- блоки вентилів (схеми І) та мультиплексори, що забезпечують виконання передачі по вибраному напрямку та в потрібний момент часу.

До групи вузлів перетворення можуть відноситися:

- суматори, що виконують у ряді випадків декілька різних мікрооперацій;
- схеми виконання логічних операцій, що іноді суміщаються з суматорами;
- схеми корекції, наприклад, для операцій десяткової арифметики;
- схеми зсуву;
- перетворювачі коду, пристрої отримання зворотних або додаткових кодів;

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

- лічильники, які використовуються для допоміжних перетворень і для підрахунку числа циклів в циклічних операціях.

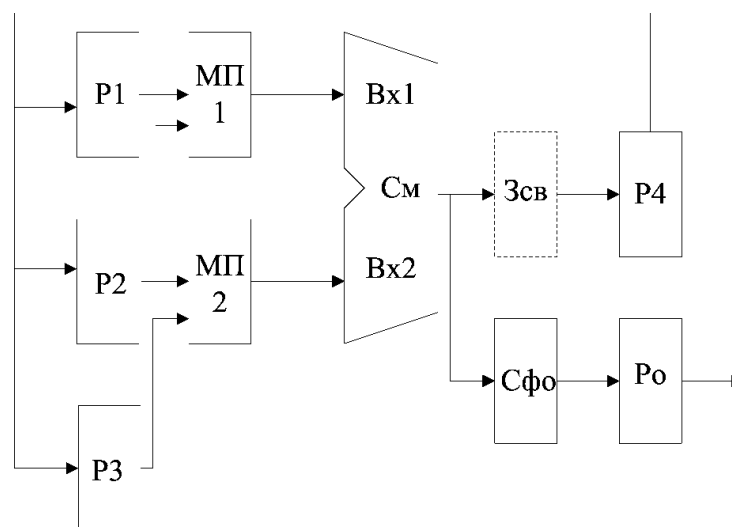
До вузлів управління можна віднести:

- блок управління АЛП (якщо такий є окремо);
- дешифратори керуючих сигналів (кодів);
- схеми формування логічних умов (ознак), які використовуються для організації галужень в мікропрограмах виконання операцій.

Звичайно, слід враховувати, що будь-яке розділення, класифікація мають елемент умовності та можна знайти достатньо прикладів, коли один вузол можна віднести до різних груп, Зокрема, в даному випадку, мультиплексори та блоки вентилів можна віднести як до вузлів передачі, так і до вузлів управління, оскільки вони дозволяють або забороняють передачу.

Арифметично-логічний пристрій включає вузли перерахованих груп, сполучені тим або іншим способом в деяку структуру. Структурні схеми АЛП можуть бути різними, що визначається відмінністю принципів їх побудови. Основні особливості організації АЛП розглянуті в класифікації цих пристроїв.

Типова структурна схема АЛП показана на рисунку 1.1, де Р1, Р2, Р3, Р4 – регістри, МП1, МП2 – мультиплексори, См – комбінаційний суматор, Зсв – зсувач, СхФормПр – схема формування ознак, РП – регістр ознак:



## Рисунок 1.1 – Типова структурна схема АЛП

У цій схемі основним вузлом перетворення інформації є суматор, що виконує операції підсумовування та логічні операції. У деяких АЛП логічні операції виконуються в спеціальних вузлах. Крім того, перетворення здійснюються в зсувачі. Регістри P1...P3 служать для зберігання операндів і проміжних результатів, регістр P4 – вихідний, використовується для проміжного зберігання результатів, що знімаються з виходу суматора (зсувача). Мультиплектори Mп1 і Mп2 забезпечують комутацію на входи суматора вмісту регістрів P1 ... P3, а в деяких випадках також інвертування, тобто отримання зворотних кодів їх вмісту (якщо в суматорі не виконується операція віднімання безпосередньо). СхФормПр забезпечує формування значень логічних умов, що відображають отримання на виході суматора нульового числа, негативного числа, переповнювання результату, перенесення із старшого розряду, парності результату та ін., а регістр ознак РП (або набір тригерів) зберігає значення цих умов.

Розглянута структурна схема має узагальнений характер. На ній не показані вузли управління, допоміжні вузли. Використання в схемі суматора комбінаційного типу обумовлює певні особливості зв'язків між вузлами АЛП. Схеми, що приводяться в деяких літературних джерелах, з суматором накопичуючого типу мають дещо інший регістровий склад і вид зв'язків. Але оскільки накопичуючий суматор зазвичай реалізується на базі комбінаційного суматора та регістра, то структура такого АЛП мало відрізнятиметься від розглянутої.

Крім набору операцій і структурної організації, АЛП характеризуються ще низкою показників. До них відносяться розрядність оброблюваних чисел (кодів), тривалість виконання різних операцій або, іноді, усереднена швидкодія, наявність додаткових функціональних можливостей типу контролю

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

правильності виконання операцій, стійкості до відмов, а також конструктивні характеристики, такі як габарити, енергоспоживання, надійність та ін.

## 1.2 Класифікація арифметично-логічних пристроїв ЕОМ

У процесорах сучасних ЕОМ використовуються різні по своїй організації АЛП. Їх відмінності обумовлені функціональним призначенням АЛП, способами реалізації операцій, вимогами по швидкодії та ін. Основні характерні особливості того або іншого АЛП можна віднести до однієї з трьох груп: особливості оброблюваної інформації, організації виконання операцій і структурної організації. Розглянемо ці групи докладніше.

Наведено загальний вид розглянутої класифікації (рисунок 1.2).

Слід зазначити, що можливий поділ АЛП за різними ознаками.

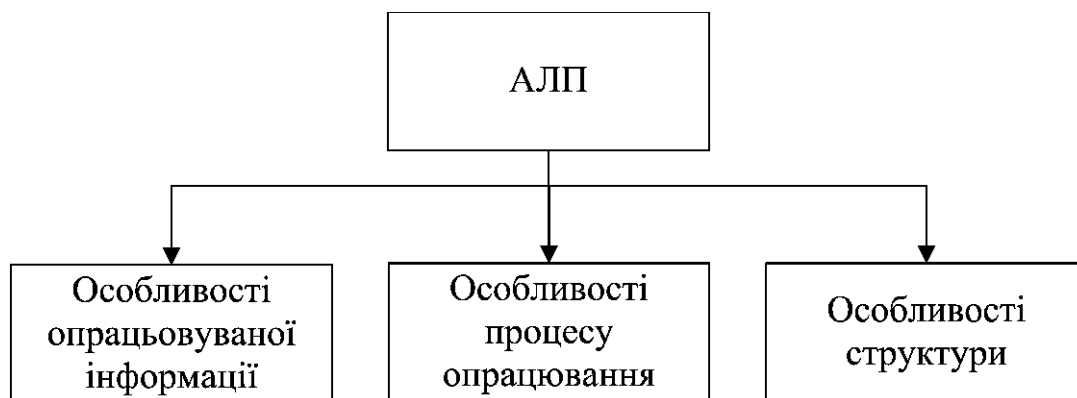


Рисунок 1.2 – Загальний вид класифікації АЛП

Оброблювана в АЛП інформація представлена або чисельною або логічною величиною (і ті та інші можуть мати різну організацію). Чисельні величини в ЕОМ представляються по-різному. Це виявляється, в основному, у використовуваних формах представлення даних, системах числення, розрядності, застосованих кодах. По цих ознаках АЛП можна розділити таким чином (рисунок 1.3).



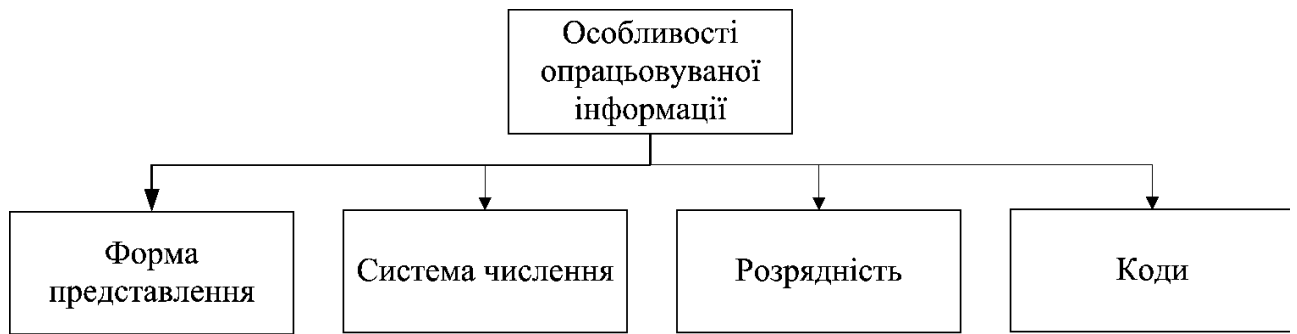


Рисунок 1.3 – Класифікація АЛП за особливістю опрацьовуваної інформації

За формою представлення чисел: АЛП з фіксованою комою; АЛП з плаваючою комою; АЛП з фіксованою та плаваючою комами (універсальні) (рисунок 1.4). Причому в рамках кожного представлення є деякі відмінності, що спричиняють за собою особливості процедур виконання операцій або структури АЛП [21]. Так, числа з фіксованою комою можуть бути представлені у вигляді цілих або у вигляді дробових чисел, менших одиниці. Це позначається на особливостях виконання операцій множення та ділення. Числа з плаваючою комою можуть мати мантису та порядок (ціле із знаком) або мантису та характеристику (зміщений порядок), що впливає на процедури обробки порядків.

Крім того, слід також згадати наявність й інших форм представлення, що використовується в калькуляторах: так звана автоматична кома, при якій положення коми в результаті операції визначається кількістю розрядів дробової частини чисел, що беруть участь в операції; природна кома – те ж для цілих частин [6].

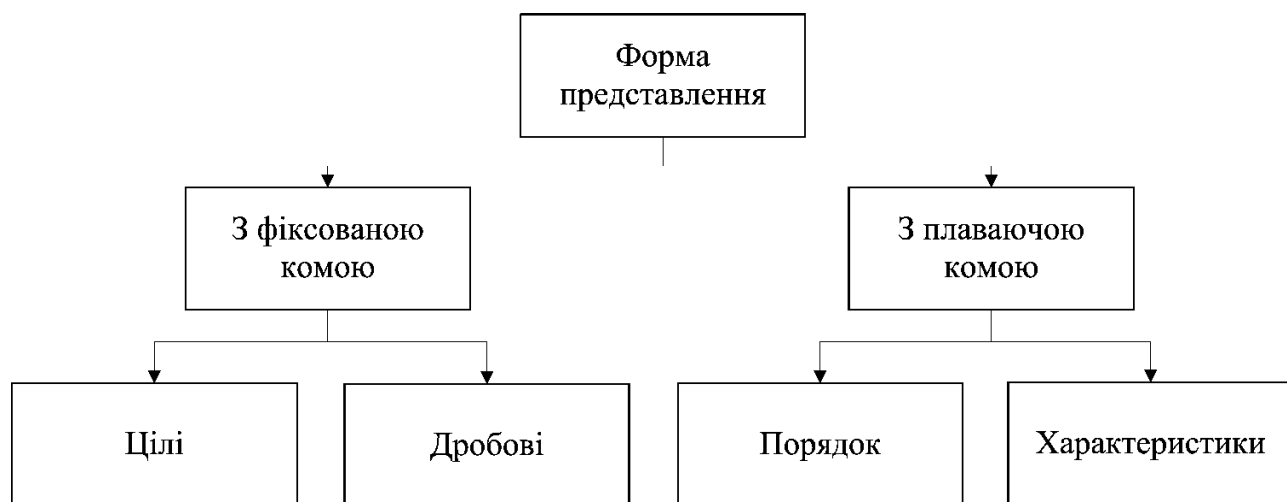


Рисунок 1.4 – Класифікація АЛП за формою представлення

За системою числення: АЛП, що працюють в позиційній системі числення; АЛП, що працюють в непозиційній системі числення (рисунок 1.5).

Відомо декілька позиційних систем числення, використовуваних в ЕОМ. В першу чергу це двійкова та двійково-десятькова системи числення. Причому остання також має різновиди, що застосовувалися в різних ЕОМ і відрізняються вагами двійкових розрядів, зсувом цифри нуль і ін. Окрім цих систем, широко поширені вісімкова та шістнадцяткова системи числення, що дають, в порівнянні з двійковою, велику наочність в зображенні чисел і що розширюють діапазон їх представлення при однаковій кількості двійкових розрядів (по відношенню до двійкової системи) в записі числа для формату з плаваючою комою.

Відомі також випадки використання трійкової системи числення, деякі роботи з R-значними системами, розробки по використанню систем числення, ваги двійкових розрядів в яких відповідають числам Фібоначчі.

З непозиційних систем числення в арифметиці використовується система залишкових класів (СЗК), числа в якій представляються у вигляді залишків від ділення початкового числа на набір взаємно-простих чисел. Таке представлення забезпечує можливість незалежної обробки розрядів (залишків) чисел, що, зокрема, представляє інтерес для цифрової оптичної обробки інформації [3].

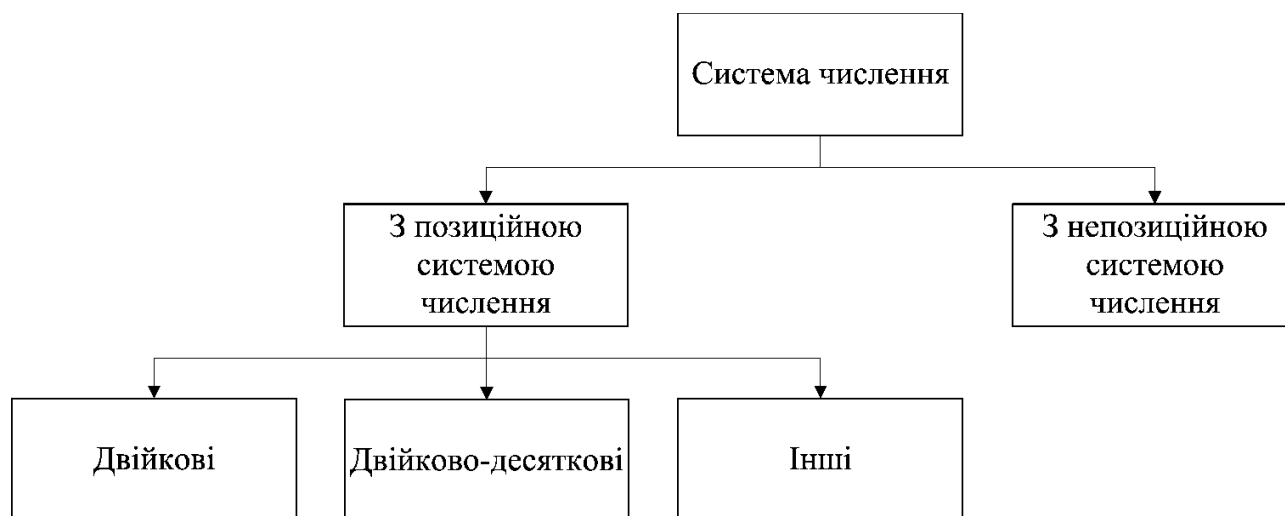


Рисунок 1.5 – Класифікація АЛП за системою числення

За розрядністю оброблюваних чисел (рисунок 1.6):

- АЛП, що виконують операції над числами (кодами) фіксованої розрядності;
- АЛП, які опрацьовують операнди змінної довжини.

У обох випадках АЛП має фіксовану розрядність блоків, але в другій групі передбачені спеціальні засоби, що забезпечують обробку операндів по частинах, та відповідні мікропрограми виконання операцій. Є також АЛП, в яких операції виконуються над декількома різними видами операндів фіксованої розрядності, звичайно це формати півслова, слова та слова подвійної довжини.

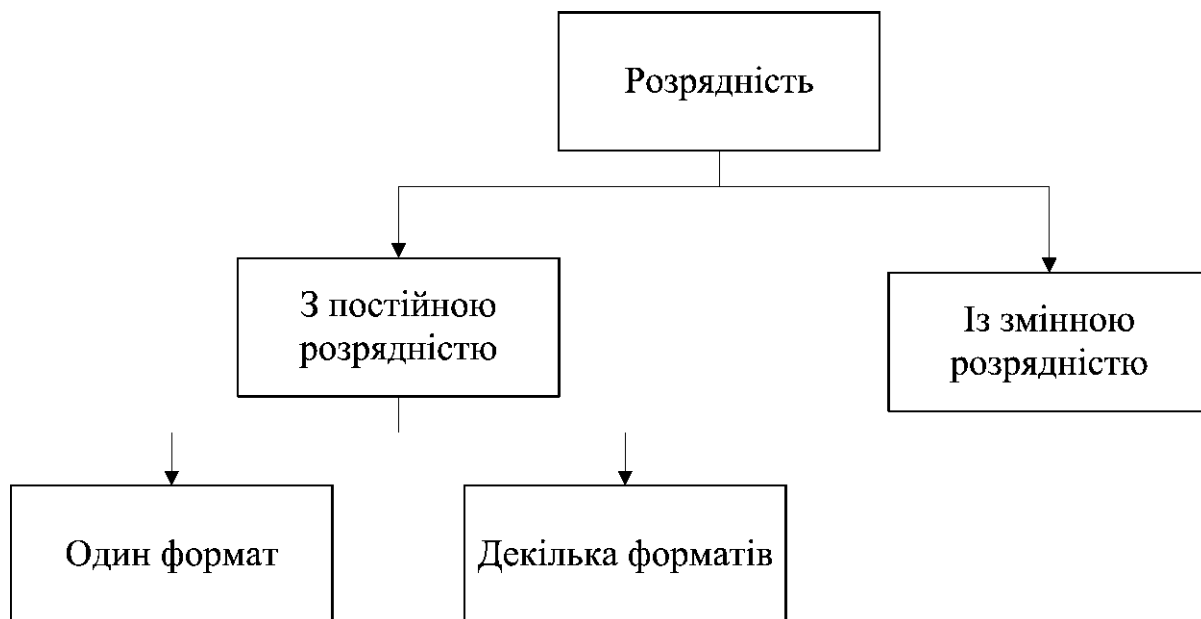


Рисунок 1.6 – Класифікація АЛП за розрядністю

За кодами, які використовуються для представлення негативних чисел (рисунок 1.7):

- АЛП з використанням зворотних кодів;
- АЛП з використанням додаткових кодів.

Принципових особливостей структур АЛП ця відмінність не обумовлює. Відомі також пристрої, в яких одні операції виконуються з використанням зворотніх кодів, а інші – додаткових.



Рисунок 1.7 – Класифікація АЛП за кодами

Особливості організації виконання операцій (процесу обробки) виявляються в принципах отримання результатів і порядку обробки даних. За цими ознаками можливе проведення таких поділів (рисунок 1.8).

За принципом отримання результату:

- АЛП з алгоритмічною реалізацією операцій;
- табличні АЛП;
- таблично-алгоритмічні АЛП.

АЛП з алгоритмічною реалізацією операцій – найбільш поширений тип. У них кожна операція (окрім найпростіших) представляється у вигляді послідовності простіших перетворень – мікрооперацій. Послідовність цих перетворень визначається алгоритмом виконання операцій. Реалізується така послідовність або за декілька тактів під управлінням відповідної мікропрограми, що забезпечує необхідну настройку вузлів АЛП в кожному такті, або за один такт, за наявності для всіх мікрооперацій окремих вузлів, сполучених в необхідній послідовності.

У табличних АЛП результат операції не обчислюється кожного разу при її виконанні. Він вибирається з таблиці – постійної пам'яті, в якій заздалегідь записані значення результатів, відповідні всім можливим значенням операндів. Такий спосіб найбільш ефективний для обчислення складних функцій одного аргументу при невеликій його розрядності, наприклад, тригонометричних функцій. Застосовується він і для реалізації звичайних арифметичних операцій.

При виконанні операцій в табличних АЛП значення аргументу (аргументів) використовують як адресу комірки ПЗП, в якому записаний результат, відповідний цьому значенню (значенням). Табличний спосіб забезпечує високу швидкість обробки, оскільки незалежно від складності перетворень, що реалізуються, всі дії зводяться до зчитування готового результату з ПЗП. Проте його недоліком є необхідність великого об'єму пам'яті (таблиці) при збільшенні розрядності операндів.

Таблично-алгоритмічні АЛП є компромісом між першими двома способами. У них результат отримують поєднанням цих способів. Частина

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

розрядів операндів (зазвичай старші розряди) використовується для отримання наближеного значення результату табличним способом. За рештою розрядів обчислюється поправка до попереднього результату. Цей метод дає можливість скоротити об'єм таблиць при збереженні високої швидкості та знайшов застосування в потужних ЕОМ.

За порядком обробки даних виділяють:

- послідовні АЛП;
- паралельні АЛП;
- конвеєрні АЛП.

Ці АЛП розрізняються між собою за ступенем паралелізму виконання операцій. Так, в АЛП послідовного типу обробка операндів здійснюється послідовно розряд за розрядом. У АЛП паралельного типу операції проводяться одночасно над всіма розрядами операндів.

Відомі також проміжні варіанти організації АЛП – паралельно-послідовні, в яких обробка операндів здійснюється одночасно за групами розрядів, тоді як групи обробляються між собою послідовно.

У АЛП конвеєрного типу паралелізм має місце на рівні операцій, тобто в них можливе виконання декількох операцій одночасно. Термін «конвеєрні АЛП» має різні інтерпретації. У ряді випадків його застосовують до багатоблокових АЛП. Наприклад, АЛП, що має як окремі блоки суматор, пристрій множення та пристрій ділення може забезпечувати конвеєрну обробку. З іншого боку, самі пристрої множення та ділення можуть бути конвеєрного типу та реалізувати відразу декілька операцій множення або ділення, які в один і той же момент часу перебувають в різних стадіях свого виконання. Ці варіанти конвеєрів називають конвеєрами послідовного типу на відміну від векторних конвеєрів [18], що виконують операції над векторами.

Очевидно, що чим вище ступінь паралелізму, закладеного в структурі АЛП, тим більше високопродуктивним він є. Проте складність схем, а отже, апаратні витрати на реалізацію таких АЛП і управління ними теж зростають.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

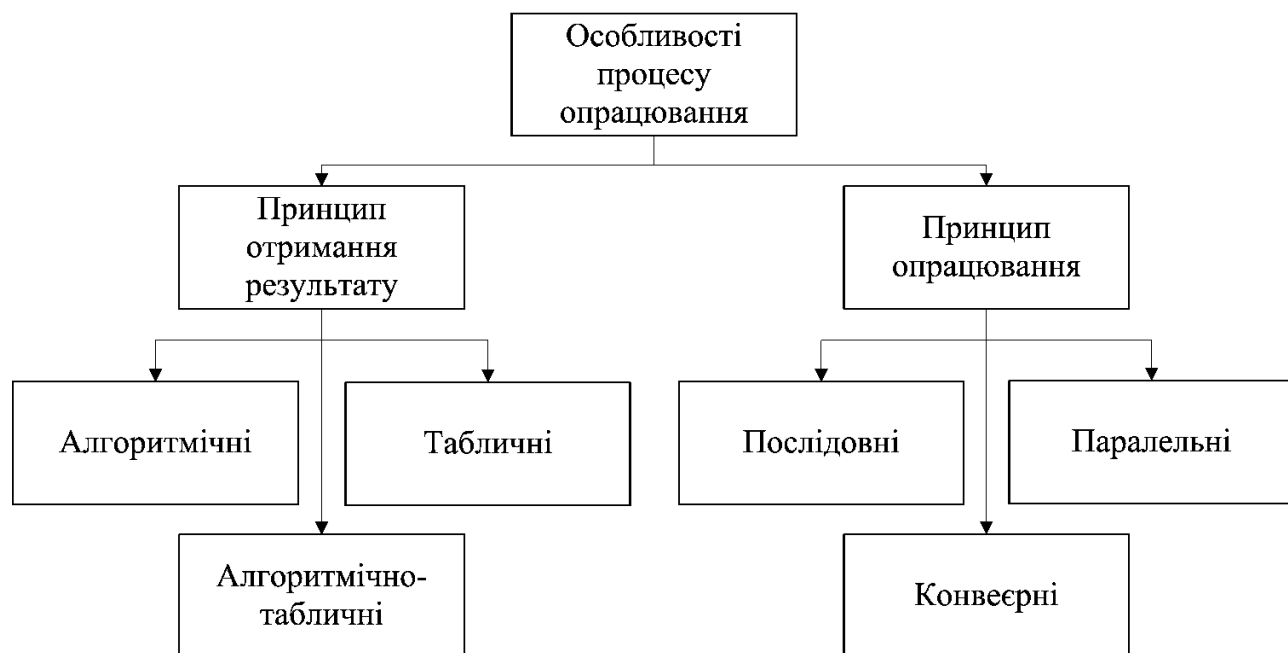


Рисунок 1.8 – Класифікація АЛП

Особливості структурної організації АЛП визначаються складом операційних блоків пристрою та характером зв'язків між ними. У цій групі ознак АЛП можна підрозділити таким чином (рисунок 1.9).

За кількістю операційних блоків:

- одноблокові АЛП (інакше, універсальні або багатофункціональні);
- багатоблокові АЛП.

Перші з них мають один операційний блок, в якому може виконуватися будь-яка з операцій АЛП. Така організація характерна для ЕОМ невисокої продуктивності. Багатоблокові АЛП мають в своєму складі декілька операційних блоків, кожен з яких орієнтований на виконання якій - або одній операції, наприклад множення, або декількох операцій, наприклад додавання та логіки. Причому передбачається одночасна робота різних блоків, що, спільно із спеціалізацією блоків, забезпечує вищу продуктивність ЕОМ з такими АЛП. По характеру зв'язків: пристрої з магістральними та з безпосередніми зв'язками.

Для перших з них характерна наявність внутрішньої шини даних, по якій здійснюються всі передачі інформації між будь-якими вузлами АЛП. У разі

безпосередніх зв'язків в структурі передбачається набір індивідуальних шин, що зв'язують пари вузлів, між якими повинні виконуватися передачі.

Структурні особливості визначаються також і призначенням ЕОМ, до складу яких входить АЛП, в цілому. Так, наприклад, в спеціалізованих системах для обробки сигналів операційні блоки можуть мати структуру, найбільш пристосовану до алгоритмів обробки [23].

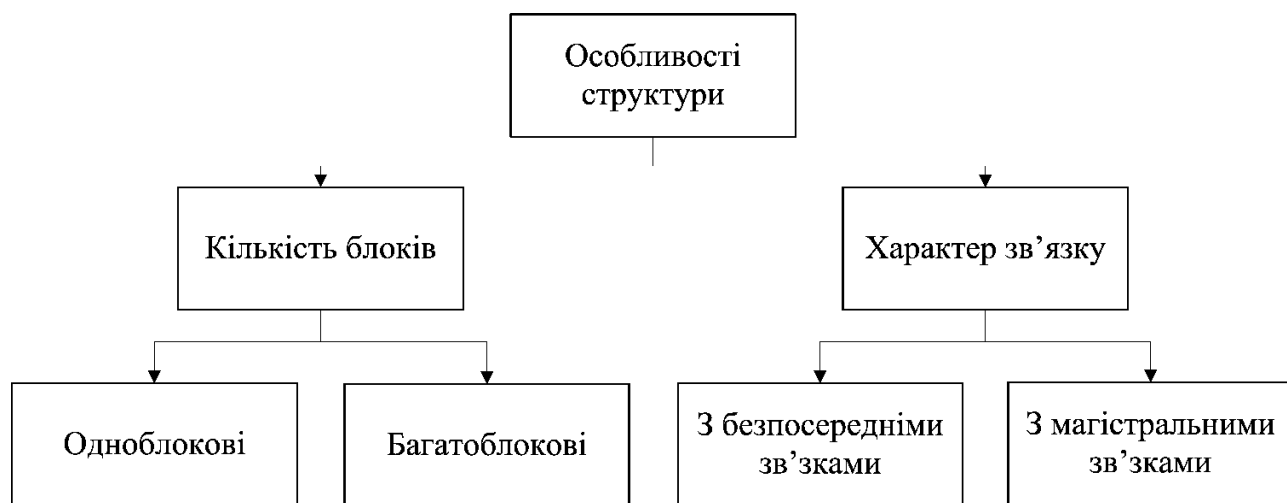


Рисунок 1.9 – Класифікація АЛП



## 2 ЗАСОБИ ПРЕДСТАВЛЕННЯ АРИФМЕТИЧНО-ЛОГІЧНИХ ПРИСТРОЇВ ЕОМ

Необхідність в представленні АЛП виникає в таких випадках:

- а) при описі пристроїв в документації;
- б) при їх проектуванні, як ручному, так і автоматизованому;
- в) при моделюванні функціонування з метою аналізу характеристик і перевірки правильності роботи.

При цьому, оскільки ЕОМ в цілому та АЛП можна віднести до класу багаторівневих систем, їх представлення може здійснюватися на різних рівнях деталізації: починаючи від рівня електронних компонентів логічних елементів і закінчуючи рівнем всього пристрою в цілому. Крім того, представлення АЛП може бути орієнтоване на опис структури пристрою, його функціонування та його технічної реалізації. Тому для різних цілей і рівнів представлення АЛП повинні існувати різні формальні та мовні засоби (крім цього, дещо самостійною, але тісно пов'язаною з ними областю є описи процесів проектування цифрових пристроїв).

Початок систематичної розробки засобів представлення ЕОМ прийнято відносити до 60-х років ХХ сторіччя, коли почали з'являтися мови для опису апаратних засобів ЕОМ. Ці мови появились після розробки мов програмування. Але зрозуміло, що засоби представлення ЕОМ існували і на раніших періодах розвитку обчислювальної техніки. Вони запозичувалися як з математичних, так і з інженерних (електротехнічних, радіотехнічних, зв'язкових) дисциплін.

З математичних дисциплін були запозичені апарат булевих функцій і алгоритмічні представлення процесів обчислень. З інженерних дисциплін були використані представлення у вигляді електричних схем, блок-схем.

На перших порах, коли обчислювальні машини були відносно прості, а проектування їх велося виключно вручну, такого представлення було досить.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Проте з ускладненням самих ЕОМ, збільшенням числа рівнів ієрархії в них, розвитком методів їх проектування та теоретичних розробок знадобилися придатніші для машинного використання формалізовані методи опису ЕОМ. В результаті створено різні формальні моделі та мовні засоби, орієнтовані на різні рівні представлення та цільове призначення. В більшості випадків ці засоби не були вузькоспеціалізованими та значна частина з них придатна для використання на різних рівнях і для різних цілей.

При представленні ЕОМ і систем залежно від ступеня деталізації розрізняють від чотирьох до семи та більше рівнів. Один з можливих варіантів містить шість рівнів представлення структури та відповідні їм процеси функціонування та технічну реалізацію, а також засоби представлення (таблиця 2.1). Багато засобів, перерахованих в таблиці, не потребують особливих пояснень. Необхідно привести лише деякі уточнення, що стосуються мов опису. Ці мови можна розділити на три групи, відповідні різним аспектам опису обчислювальних засобів: мови структурного опису, мови функціонального опису та мови графічного опису [13].

Таблиця 2.1 – Рівні та засоби представлення обчислювальних пристроїв

Рівень представлення структури	Процеси функціонування	Технічна реалізація	Засоби представлення структури	Засоби представлення функціонування	Засоби представлення реалізації
1. Електронні схеми	Струми в електричних колах	Радіо-компоненти	Електрична схема (принципова)	Диференційні рівняння для струмів і напруг	Фотошаблони масок інтегральних схем (мови графічного типу)
2. Логічні схеми	Логічні перетворення, перемикання	Інтегральні схеми	Схема з логічних елементів	Булеві функції, цифрові автомати	Топологія інтегральних схем (мови

	станів елементів пам'яті		(електрична принципова )		графічного типу)
3. Вузли та блоки	виконання мікрооперацій	Інтегральні схеми, конструктиви (плата, ТЕЗ), їх фрагменти	Схеми з логічних елементів і вузлів (електрична функціональна )	Мови регістрових передач (мікрооперацій)	Плани кристала (мови графічного типу)
4. Пристрої	Виконання операцій (мікропрограм)	Інтегральні схеми, плати, Тези, панелі, стійкі	Схеми з вузлів і блоків (електричні структурні)	Мови регістрових передач, мови опису мікропрограм	Укрупнений план кристала, креслення конструктиву
5. ЕОМ	Виконання команд (програм)	Конструктиви всіх рівнів	Структурні схеми	Мови команд, мови програмування	Креслення конструктиву
6. Системи	Взаємодія пристроїв, обчислювальний процес	Конструктиви всіх рівнів	Структурні схеми. Для всіх рівнів : Мови структурного опису	Мови моделювання систем, мережеві моделі	Креслення конструктивів

Мови першої групи описують, головним чином, з'єднання між елементами (вузлами) структури. Достатньо часто самі елементи структури розглядаються при цьому як «чорні скриньки», що мають входи та виходи. Причому в розвинених мовах цієї групи зазвичай закладена можливість багаторівневого ієрархічного опису структури, коли на вищих рівнях елементами структури є підструктури, опис яких побудований раніше. Прикладами мов цієї групи можуть служити мови СТРУКТУРА [2, 5], SDL, HSL [13] та інші.

Мови функціонального опису описують розповсюдження сигналу, функцію елемента або блоку, алгоритм функціонування, зміни стану. Якщо

						Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата	Ошибка! Источник ссылки не найден. ПЗ	

структурні описи відображають статичні зв'язки на рівні структури, то функціональні описи представляють динаміку роботи пристроїв, послідовність процесів, що в них відбуваються. На відміну від мов структурного опису, які легко інтегруються в ієрархічну структуру, опис функціонування на різних рівнях спричиняє за собою значні відмінності у використовуваних моделях, а отже, і в мовах представлення.

На найвищому рівні (системному або архітектурному) використовується імовірнісний опис взаємодії різних пристроїв, наприклад, моделі теорії масового обслуговування, мережі Петрі. Такому рівню відповідають мови опису імітаційних моделей типу GPSS, СИМСКРИТ, СИМУЛА [19] та інші.

Таким є рівень програм і машинних команд, якому відповідають мови програмування та мови машинних команд.

Ще нижчим рівнем функціонального опису є рівень мікрооперації та мікропрограм, на якому функціонування описується в термінах переходів реєстрів і тригерів з одного стану в інший. Для цього використовуються мови представлення мікропрограм типу мов АЛГОРИТМ [8], МИКРОПРОГРАМА [2] або мови реєстрових передач CDL, DDL, HSL-FX та інші [13].

Іноді для представлення АЛП та інших дискретних пристроїв використовують мови, що дають можливість описувати структуру та функціонування одночасно, наприклад мова ДИСТАЛ (Дискретні Структури та Алгоритми) [12]. Можна додавати такі описи на алгоритмічних мовах типу АPL, Паскаль і Сі, проте такі описи можуть бути не наочними та громіздкими.

Нарешті, для опису технічної реалізації обчислювальних пристроїв використовуються графічні мови. За допомогою цих мов можуть описуватися маски та топологія інтегральних схем, зображатися фотошаблони для виготовлення друкованих плат, задаватися конфігурації радіокомпонентів і мікросхем, зображатися контури друкованих плат, розміщення на них елементів, будуватися символічні зображення елементів і т.д. Слід враховувати, що власне графічних мов небагато [12]: це, наприклад, мови CIF і SMF. Частіше виникає потреба в побудові та обробці файлів графічних зображень за допомогою

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

спеціальних програмних пакетів або графічних програмних засобів, передбачених в тій або іншій системі автоматизованого проектування.

Багато прикладів використання мов різних типів приведено в [12].

## 2.1 Узагальнені функції систем переробки інформації

Людську діяльність можна розглядати як процес постановки людиною деяких цілей і виконання відповідних перетворень матеріального та ідеального, що ведуть до поставлених цілей. В даний час в різних дослідженнях, присвячених питанням проектування складних технічних систем, як об'єкти цих перетворень прийнято розглядати речовину, енергію або інформацію.

Типи основних системних процесів: перетворення, зберігання, передачі та управління. Для систем опрацювання інформації (СОІ) об'єктом процесів є інформація. Проте будь-який поділ не є абсолютним, та в будь-якому перетворенні беруть участь, в тій чи іншій мірі, всі три названі категорії. Прикладом є те, що інформація повинна представлятися яким-небудь речовим або енергетичним носієм.

Тому при проектуванні СОІ питання організації подачі електроживлення, відведення теплоти, екранування від електромагнітного випромінювання (процеси, об'єктом яких є енергія), а також механічних переміщень носіїв інформації та конструктивного оформлення (об'єктом яких є речовина), що безпосередньо виникають на етапі технічного проектування, не можуть випускатися з уваги повністю та на попередніх етапах створення систем. Проте, в роботі вся увага зосереджена переважно на інформаційній стороні тих, що протікають в СОІ процесах.

Кожен з чотирьох названих типів процесів дає можливість виділити різні аспекти функціонування СОІ та сформулювати відповідну групу вимог при розробці системи, сукупність яких і направлятиме процедури формування

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

структури систем. Так, процеси зберігання (процеси М-типу) пов'язані із забезпеченням «існування» в системі інформаційних об'єктів: констант, змінних, масивів, файлів та інших і визначають вимоги до складу, функціональних і технічних характеристик засобів запам'ятовування інформації всіх рівнів, починаючи від тригерів стану та закінчуючи зовнішньою пам'яттю.

Процеси перетворення (процеси Р-типу) пов'язані з видозміною існуючих інформаційних об'єктів або побудовою нових і визначають вимоги до складу, функціональних можливостей і технічних характеристик операційних елементів, блоків і пристроїв СОІ.

Процеси передачі (процеси Т-типу) пов'язані із забезпеченням єдності всієї системи, що складається з операційних запам'ятовувальних вузлів, дозволяючи переміщати інформаційні об'єкти до вузлів їх обробки та зберігання, а також в керуючу частину (для аналізу). Характер цих процесів визначає склад, функції та параметри ліній зв'язку всіх рангів від простого з'єднання входів і виходів елементів до інтерфейсів мереж ЕОМ.

Процеси управління (процеси С-типу) пов'язані з організацією решти всіх процесів і є узагальненою формою тієї сукупності процесів, якою вони управляють. Саме тому процеси С-типу реалізуються на основі процесів М-, Р- і Т-типів простішого вигляду, ніж ті процеси, управління якими вони здійснюють. С-процеси визначають вимоги до функціональних можливостей, організації та алгоритмів управляючої частини СОІ.

Використаний вище для визначення людської діяльності термін «перетворення» є достатньо загальним і може застосовуватися для позначення різних дій: арифметичних операцій, вирішення систем рівнянь, перетворення енергії. Тому для перетворення інформації доцільно дещо конкретизувати трактування цього поняття. Такою обмеженою, але достатньо широкою інтерпретацією перетворень інформаційних об'єктів є відображення множин, зокрема множин, із заданими на них співвідношеннями. Розглядаючи перетворення як відображення, позначатимемо їх в загальному випадку

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

символом F, підкреслюючи їх близькість в такому трактуванні до поняття функції.

Реалізація перетворень в системі переробки інформації здійснюється на основі взаємодії вузлів системи, які можна розглядати як елементи її структури. У першому розділі наголошувалося, що рівень такого розгляду може бути різний залежно від його цілей і завдань: від p-n переходів на кристалі ВІС до цілих систем ЕОМ. Проте при вирішенні завдань проектування ЕОМ, пов'язаних з розробкою структури окремих її пристроїв, зокрема арифметико-логічного пристрою, аналіз структури здійснюється, як правило, на рівні регістрових передач і зазвичай не опускається нижче за рівень логічних елементів.

При необхідності побудувати структуру, що реалізує деяке перетворення F, слід представити її у вигляді зв'язної сукупності відомих структур (елементів). Для цього само перетворення F повинне бути розбито (проведена декомпозиція) на зв'язну сукупність таких перетворень, виконання яких можливо здійснити на відомих структурах (елементах). Враховуючи названий вище рівень розгляду структур пристроїв ЕОМ, до таких перетворень можуть бути віднесені перетворення (таблиця 2.2). У цій таблиці кожному функціональному перетворенню поставлений у відповідність структурний елемент СОІ, що реалізує це перетворення, а також вказаний його тип: к – комбінаційний, н – накопичуючий.

Таблиця 2.2 – Основні функції Р-типу рівня регістрових передач

Перетворення	Структурний елемент	Тип елемента
Підсумовування двох код/бітів	Суматор	К/Н
Додавання одиниці (лічба)	Лічильник	К/Н
Віднімання одиниці	Лічильник	К/Н
Зсув коду/біта	Зсувний регістр Зсувач	Н К
Порівняння код/бітів (рівність, більше, менше)	Схема порівняння Суматор	К К/Н
Порівняння з константою	Схема аналізу вмісту	К
Логічні функції: І, АБО,	Схема логічних операцій	К/Н

ВИКЛЮЧЕ АБО, НІ	(суматор)	
Довільна логічна функція	Комбінаційна схема, ПЛМ	К
Шифрування (перетворення коду)	Шифратор	К
Дешифрування	Дешифратор	К

До функцій Т, М і С-типів на даному рівні регістрових передач належать дії (таблиця 2.3). У графі 3 цієї таблиці показано, що розділення процесів і функцій на Р, С, М і Т типи є відносним, не абсолютним. Кожен тип процесу в тій чи іншій мірі поєднується з іншими або навіть може бути замінений ними. Наприклад, перетворення неможливі без подачі інформації на входи вузла перетворення та знімання результату з його виходу, умову для управління умовним переходом часто доводиться обчислювати та т.п. Прикладом заміни типів перетворень може служити табличне АЛП, в якому Р процеси замінені на М-тип. Іншим прикладом є дублювання обчислень значення однієї та тієї ж величини в різних блоках замість обчислення її в одному блоці з подальшою передачею в іншій. Такі взаємозв'язки, зокрема, обумовлюють, разом з іншими причинами, різноманітність структур СОІ і неоднозначність проектних рішень

Таблиця 2.3 – Основних функції Т-, М- і С- типу рівня регістрових передач

Функція	Функціональний елемент, блок	Примітка
1	2	3
<b>Т-тип</b>		
Подача коди /біта з джерела (вузла, шини ) на вхід вузла	Жорсткий зв'язок (виходу вузла з джерелом). Зв'язок (входу вузла з джерелом) через блок вентилів, мультиплексор	Т+С
Подача фіксованої коди/біта на вхід вузла	Жорсткий зв'язок інформаційного входу вузла з 0/1. Шина управління інформаційним/синхровходом вузла	Т=С
Видача коди/біта з виходу вузла ( у приймач : вузол, шину)	Жорсткий зв'язок виходу вузла з приймачем. Зв'язок виходу вузла з приймачем через блок вентилів, мультиплексор	Т+С



<b>М-тип</b>		
Зберігання коди/біта	Тригер, регістр, блок регістрів, ЗУ	
Фіксація (запам'ятовування) коди/біта	Тригер, регістр, блок регістрів, ЗУ	М+С
Зберігання фіксованого значення коди/біта	Шина, що управляє	
<b>С-тип</b>		
Дозвіл виконання передачі, перетворення (синхронізація)	Керуючий автомат (вхід)	
Завдання виконуваних функцій	Керуюча шина Керуючий (кодовий вхід)	С+Р (Дш)
Безумовний перехід до такої дії	Керуючий автомат (генератор синхроімпульсів)	
Умовний перехід до наступної дії	Керуючий автомат	С+Р

## 2.2 Загальна методологія та етапи проектування АЛП

В даний час існує велика різноманітність методів вирішення завдань проектування систем різного призначення. Серед них є як методи, призначені для вирішення певних завдань, так і комплексні методики побудови тих або інших систем.

Однією з методологій, що найповніше охоплює всі аспекти проектування, є еволюційний синтез систем, що базується на функціонально-структурному підході до аналізу та синтезу систем. Цим підходом є сукупність концепцій, об'єктивних закономірностей розвитку, положень і висновків, що визначають стратегію аналізу та синтезу [5].

Згідно з функціонально-структурним підходом при проектуванні СОІ, яка призначена для виконання певних функціональних перетворень, здійснюють перехід від функцій системи переробки інформації до її структури.

Характерними особливостями функціонально-структурного підходу (ФСП) є:

- облік взаємозв'язку функцій і структури об'єктів при визначальній ролі функцій по відношенню до структури;

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

- цілісність і спільність підходу до аналізу та синтезу багаторівневих систем;
- облік енергетичних та інформаційних зв'язків між елементами системи;
- сумісне використання загальних законів матеріального світу та закономірностей розвитку систем певного класу;
- розгляд систем в розвитку.

Початковою посилкою ФСП є єдність функцій і структури системи при визначальній ролі функції. Підхід базується на концепції інтеграції та диференціації функції в процесі розвитку – концепції багатofункціональності та спеціалізації елементів, підсистем і систем. Цілісність підходу до аналізу та синтезу систем забезпечується формуванням спеціальної моделі – «дерева функцій», що є декомпозицією цільової функції системи. В процесі багаторівневої декомпозиції формуються основні та додаткові функції, що реалізуються окремими підсистемами та елементами, які знаходяться на різних рівнях системи. Важливою відмітною особливістю підходу є сумісний облік при аналізі та синтезі речових, енергетичних та інформаційних потоків усередині системи в процесі обміну із зовнішнім середовищем.

Аналіз розвитку систем з погляду функціонально-структурного підходу приводить до таких висновків:

1. Структура системи визначається сукупністю функцій, що реалізуються.
2. Між функціями, що реалізуються, та структурою системи не існує взаємно однозначної відповідності. Один і той же набір функцій може бути реалізований різними сукупностями багатofункціональних і спеціалізованих елементів, об'єднаних в систему.
3. При зміні умов існування системи відбувається зміна спектру функцій, що реалізуються системою, що приводить до відповідних змін в структурі системи. Таким чином, здійснюється адаптація організації системи до умов існування, що змінюються.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

4. В процесі розвитку систем формуються різні класи та види систем, функціонально-структурна організація яких найбільш адекватна умовам існування.

Методологія проектування систем, що базується на ФСП, отримала назву еволюційного синтезу систем (ЕСС). ЕСС є багатофазним процесом, що зводиться до послідовного формування та перетворень ряду моделей системи, відповідних різним рівням її організації [5]. Характерною особливістю ЕСС є широке використання на етапах аналізу систем - прототипів, формування концепції системи та синтезу її структури таких моделей:

- дерева функцій системи;
- дерева суперечностей системи;
- багаторівневої структури системи на основі функціональних модулів; структури системи на основі конструктивних модулів.

Основні фази ЕСС включають по декілька етапів - процедур і зводяться до такого:

1. Аналіз систем прототипів (виявлення функцій систем, побудова узагальненого дерева функцій, визначення базових структур і аналіз принципів реалізації).

2. Дослідження дерева суперечностей системи (аналіз «вузьких місць» прототипів, обмежуючих чинників, виявлення основної суперечності, побудова дерева суперечностей і його аналіз).

3. Формування концепції системи (виявлення способів подолання суперечностей системи, пошук альтернативи реалізації, розробка технічного завдання, визначення сукупності показників якості та критерію оцінки системи).

4. Формування дерева функцій системи (визначення множини основних і додаткових функцій, визначення числа рівнів декомпозиції, декомпозиція функцій системи, виділення типових операторів, відображення функцій на множина операторів).

5. Формування функціональної структури системи (аналіз методів апаратної та програмної реалізації, розробка алгоритмів функціонування

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

системи, аналіз зв'язків між операторами різних рівнів, побудова діаграм активності операторів, визначення завантаження ресурсів, формування функціональних модулів і їх зв'язків, виділення типових функціональних підсистем).

6. Формування технічної структури системи (вибирання технічних засобів для реалізації системи, формування таблиць відповідності функціональних і конструктивних модулів, виявлення необхідності розробки оригінальних засобів, перетворення елементів функціональної структури, покриття функціональних підсистем конструктивними модулями, формування конструктивних модулів високого рівня).

7. Оцінка показників якості та вибір остаточного варіанту системи по сформованому критерію.

Таким чином, формування структури системи здійснюються на основі принципів декомпозиції та композиції функцій і структур різних підсистем, а перетворення моделей системи проводиться в такій послідовності:

- 1) формування дерева функцій системи;
- 2) декомпозиція функцій системи до рівня прийнятого базового набору операторів (перетворення 1 і 2 виконуються ітеративно);
- 3) формування функціональної структури системи;
- 4) формування технічної структури системи.

Відомі декілька інших варіантів виділення та іменування етапів процесу проектування. Так, наприклад, приведені вище етапи 1...3 відповідають етапу системного проектування [17], етапи 4 і 5 – алгоритмічному та логічному етапам проектування [20] і т.д.

При розгляді певного класу систем або підсистем, як це має місце в даній роботі, присвяченій арифметично-логічним пристроям, виявляється можливим дещо обмежити та уточнити зміст ряду названих етапів. Крім того, багато з вхідних в них процедур може бути проведені заздалегідь, безвідносно до конкретної системи, та надалі використовуватися при проектуванні різних екземплярів систем даного класу. Це, наприклад, справедливо стосовно етапу

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

аналізу систем-прототипів, на якому виявляються функції системи та будується її узагальнене дерево функцій. Оскільки дерево функцій системи певного класу в узагальненому варіанті не повинне включати функції, представлені у вигляді конкретних дій над заданими об'єктами, то таке узагальнене дерево функцій АЛП може бути побудоване заздалегідь і використовуватися при розробці АЛП для різних систем переробки інформації. Наведено фрагмент одного з можливих варіантів узагальненого дерева функцій АЛП (рисунок 2.1), де як функції різних рівнів виділені такі:

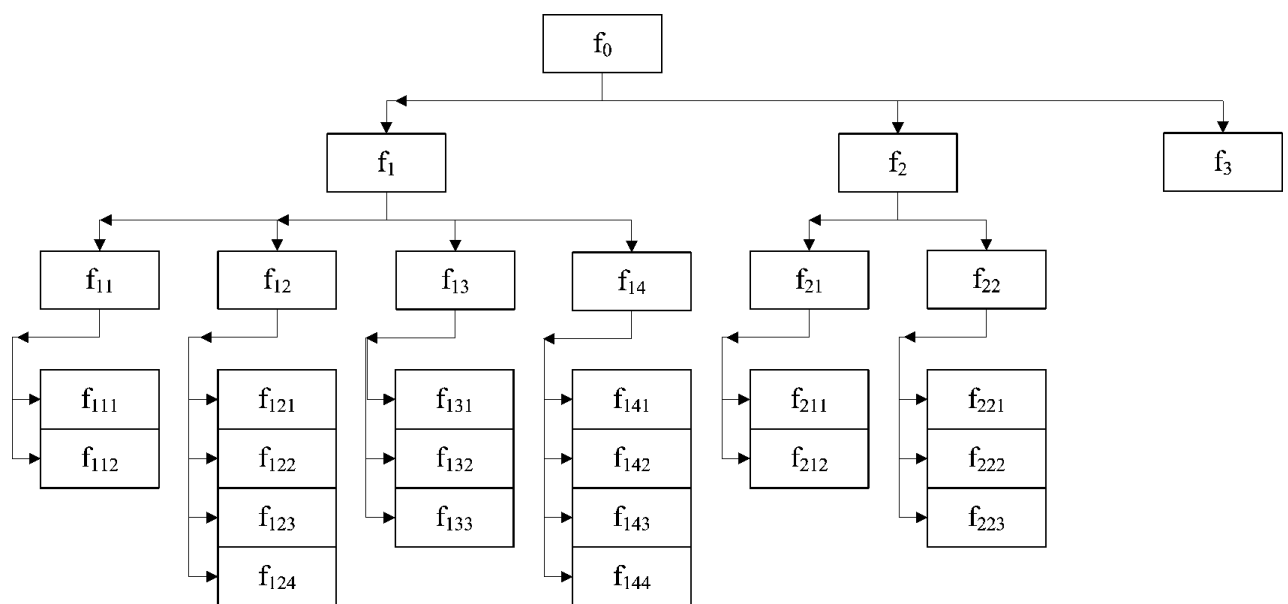


Рисунок 2.1 – Варіант узагальненого дерева функцій АЛП

Рівень 0:

- $f_0$  – виконання заданої сукупності перетворень;

Рівень 1:

- $f_1$  – формування результатів перетворень (основна функція);
- $f_2$  – забезпечення взаємодії з іншими пристроями (додаткова функція);
- $f_3$  – виконання контролю перетворень (додаткова функція);

Рівень 2:

- $f_{11}$  – обмін операндами та результатами усередині АЛП;
- $f_{12}$  – зберігання операндів і результатів;
- $f_{13}$  – перетворення операндів;

- f14 – управління формуванням результатів;
- f21 – обмін даними з іншими пристроями;
- f22 – обмін керуючою інформацією з іншими пристроями;

Рівень 3:

- f111 – подача (проміжних) операндів на блоки перетворень;
- f112 – прийом (проміжних) результатів з блоків перетворень;
- f121 – зберігання операндів;
- f122 – зберігання проміжних результатів;
- f123 – зберігання результатів;
- f124 – зберігання ознак результатів;
- f131 – перетворення коду;
- f132 – зсув чисел;
- f133 – арифметичні перетворення;
- f134 – логічні перетворення;
- f141 – формування керуючих сигналів для настройки блоків перетворення;
- f142 – формування керуючих сигналів для передачі даних між блоками;
- f143 – формування ознак результатів;
- f144 – формування ознак зайнятості/готовності АЛП;
- f211 – прийом операндів;
- f212 – видача результатів;
- f221 – прийом коди операції;
- f222 – видача ознаки зайнятості/готовності АЛП;
- f223 – видача ознак результату.

Не показано декомпозицію по рівнях функції: f3 – контроль результатів перетворень, подібна до декомпозиції основної функції f1 (див. рисунок 2.1).

Необхідно враховувати, що розбиття на функції різних рівнів, як і формулювання змісту самих функцій можуть бути різні. Деякі функції, наприклад, формування ознак результату, можуть відноситися до різних функцій більш високого рівня. Тому можлива побудова різних варіантів

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

узагальнених дерев функцій АЛП, проте змістовний сенс вхідних в них функцій, загалом, буде однаковим.

Аналогічно для етапу дослідження дерева суперечностей може бути заздалегідь виконаний аналіз «вузьких місць» прототипів, побудова та аналіз самого дерева суперечностей. Причому воно включатиме суперечності між основними показниками, якими характеризуються функції різних рівнів дерева функцій системи.

При формуванні концепції системи використовують загальні принципи вирішення протиріч, які будуть розглянуті далі.

На шостому етапі формування таблиць відповідності конструктивних і функціональних модулів також може здійснюватися, в основному, заздалегідь, особливо у разі автоматизованого проектування. Проте ці таблиці обов'язково поповнюватимуться згодом.

Крім того, особливості щодо вузького класу перетворень, що покладаються на АЛП (арифметичні та логічні операції), обумовлюють і деякі зміни змісту окремих етапів. Зокрема, на етапі формування концепції системи приймаються рішення про вибір табличної або алгоритмічної реалізації перетворень в АЛП, про необхідність підвищення надійності за рахунок включення у функції, що реалізуються, функцій контролю, про максимально можливе розпаралелювання виконуваних дій або, навпаки, про максимальне поєднання функцій в різних блоках і т.д. Останнє значною мірою визначає порядок проведення таких етапів розробки.

Необхідно також вказати, що етапи побудови дерева функцій і формування функціональної та технічної структури можна виконувати ітеративно, уточнюючи рівень декомпозиції функцій у міру формування структур. Причому безпосереднє отримання оптимальних по заданому критерію структурних рішень не завжди можливо. Тому найдоцільніше формувати структуру спочатку з максимальним розподілом функцій, яку при необхідності потім перетворити в структуру з бажаним ступенем поєднання функцій. Такий шлях і буде вибраний як базовий в подальшому викладі. Проте раніше слід

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

розглянути особливості та можливі процедури декомпозиції функцій при побудові дерева функцій АЛП.

### 2.2.1. Декомпозиції функцій АЛП

При формуванні дерева функцій системи на основі аналізу систем-прототипів відповідно до вимог технічного завдання визначається множина основних і додаткових функцій системи. Множина основних функцій визначає клас системи [5]. Множина додаткових функцій визначає такі якості системи, як надійність, живучість, сервіс, що представляється користувачам, і т.д. Зміна множини додаткових функцій породжує різні варіанти систем даного класу, які визначаються множиною основних функцій. Дерево функцій системи формується потім в результаті декомпозиції отриманої множини основних і додаткових функцій.

Декомпозиція функції  $f$  припускає розділення її на деякі складові  $f_{i1}, \dots, f_{in}$ , сумісне (одночасне або по чергове) виконання яких дає в сукупності реалізацію початкової функції  $f_i$ . Для визначення способів декомпозиції функцій необхідно знайти компоненти, що входять в опис функцій, виявити структуру, типи та оточення цих компонентів.

У загальному випадку функціями є дія (дії) над деякими об'єктами. Такі об'єкти для зручності посилання далі називатимуться актантами. Тому в опис функції обов'язково входять власне дії (дія) і актант. Окрім цього, в описі може враховуватися об'єкт, що здійснює дію, – актор, та об'єкт, через якого здійснюється дія – засіб або інструмент. Також може враховуватися зовнішнє оточення функції; оточення актанта (і актора із засобом), тобто середовище, так і зовнішня дія, тобто оточення самої дії, активність зовнішнього середовища. Таким чином, опис функцій може мати такий загальний вигляд:

*Дія, Актант [ , Актор] [ , Засіб] [ , Середовище] [ , Активність].*

Декомпозиція, представлена такими складовими функціями, може проводитися по кожній з них (за типом, структурою або іншими атрибутами). Проте найчастіше розклад проводиться за дією та за актантом.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		



Власне дія може бути піддана декомпозиції:

- за фазами: ініціація, збільшення (наростання), підтримка, зменшення (ослаблення);
- за типами: перетворення, зберігання, передача, управління;
- за класами дій в рамках ширшої множини класів дій;
- за частинами середовища дії, якими можуть бути час, простір, структура та ін.

Прикладами подібних способів декомпозиції є:

а) передача байта по послідовному інтерфейсу, передача стартових і стопових бітів початкової та кінцевої фаз або виконання перетворення, що включає подачу операндів на входи блоку та прийом результату з виходу блоку перетворення як початкову та кінцеву фази;

б) формування суми  $n$  доданків, що реалізується почерговим додаванням доданків до збереженої суми за допомогою подачі їх на вхід блоку додавання, та управління цим процесом;

в) розділення виконуваних в АЛП операцій перетворення на арифметичні та логічні;

г) зберігання кодів протягом деякого часу може бути розділене на інтервали зберігання, що послідовно повторюються, як в напівпровідникових динамічних ОЗП.

Декомпозиція функції за об'єктом дії полягає зазвичай в розділенні дії з елементів структури цього об'єкту. Наприклад, виконання логічних операцій над довгими операндами може здійснюватися порозрядно або побайтно. Іншою можливістю є декомпозиція по розділенню класу об'єктів дії на деякі групи. Наприклад, арифметичні операції над числами, представленими в прямих кодах, та операції над числами, представленими в додаткових кодах, можна розглядати як декомпозицію функції виконання арифметичних операцій.

Аналогічним чином функції можуть бути декомпоновані за іншими складовими, що входять в їх опис.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

### 2.2.2. Способи подолання суперечностей при формуванні концепції

системи

Як зазначено вище, на етапі формування концепції системи проводиться виявлення способів подолання суперечностей проектованої системи, що зводяться до погіршення одних показників системи при спробі поліпшення інших її показників. Слід враховувати, що пошук концептуальних рішень є одним з найбільш складних завдань проектування, рекомендації яких можуть носити лише загальний характер. При розробці системи можливі два основні шляхи подолання наявних або виникаючих суперечностей:

- вибір рішення, що забезпечує оптимальне по заданому критерію співвідношення між сторонами даної суперечності (баланс показників);
- часткова або повна ліквідація (зняття) даної суперечності за рахунок переходу до іншого способу або принципу реалізації.

Перший шлях пов'язаний із зміною, як правило, кількісних характеристик системи, що розробляється. Причому ці зміни можуть зачіпати як параметри елементів системи, так і кількісні характеристики її структури. Типовими прикладами такого шляху можуть служити:

- а) використання більш швидкодіючих, менш енергоємних або дешевших елементів;
- б) різні структурні зміни, пов'язані з переходом від паралельної реалізації функцій до послідовної, або навпаки;
- в) заміна апаратної реалізації програмною (мікропрограмною), і навпаки.

Для другого шляху рішення зазвичай пов'язані з якісною зміною принципів організації структури системи або реалізації її функції. Завдання пошуку концепції системи відносять до винахідницьких, інтелектуальних робіт. В цьому випадку застосовуються різні прийоми, систематизація та аналіз яких проводиться у низці праць [4, 5, 12]. Проте побудова яких-небудь формальних рекомендацій по застосуванню цих прийомів є складним завданням. Деякі спроби такого роду реалізуються у вигляді експертних систем [12].

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

До ефективних прийомів, які використовуються на даному шляху, можуть бути віднесені поєднання функцій (принцип багатфункціональності), заміна типу функцій (перетворення, зберігання, передача), інверсія функцій, побудова інваріантних систем, композиція якостей та інші. Прикладом поєднання функцій є усунення за допомогою цього принципу суперечності між вартістю точного (з малим допуском) виготовлення та швидкістю інтегральних схем в технології з самосуміщенням [16], в якій сам затвор транзистора використовується як маска при створенні областей витоку та стоку. Заміна типу функцій добре ілюструється на прикладі табличної реалізації АЛП, в якому, в порівнянні із звичайним АЛП, перетворення операндів замінюється зберіганням результату. Як завжди, звичайно, слід враховувати, що існують і такі прийоми, які можна розглядати як представники як першого, так і другого шляхів.

### 2.2.3. Формування таблиць відповідності функцій, функціональних і конструктивних моделей

При формуванні функціональної та технічної структури системи на відповідних етапах (п'ятому та шостому) проводиться покриття функцій, які створюють дерево функцій цієї системи, функціональними та конструктивними модулями. Для проведення таких процедур необхідно мати:

а) бібліотеку відображень (таблиць) тих функцій, що забезпечують виконання функціональних та конструктивних модулів (з можливим розрізненням способів їх реалізації);

б) способи вибору сукупності модулів, що реалізують необхідну сукупність функцій, оскільки зазвичай є деяка множина таких сукупностей модулів;

в) критерії оцінки переваги того або іншого варіанту.

Таблиці відповідності спочатку формуються на підставі аналізу дерев функцій і реалізації типових систем даного класу. В них для кожної функції вказуються:

– ім'я функції та її об'єкту (актанта);

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

- перелік можливих її параметрів, що відносяться до різних складових опису функції (параметрам за допомогою посилань може бути наданий вигляду дерева, а сам їх перелік можна представити окремими файлами);
- ім'я функціонального блоку, що реалізує функцію, при необхідності доповнюване особливостями блоку;
- спосіб реалізації функцій функціональним блоком, який залежно від рівня (складності) функції задається або вказівкою функцій нижчого рівня, або вказівкою структурних особливостей блоку;
- ім'я файлу конструктивних блоків, що реалізують функцію, якщо такі є. Для кожного з них повинні бути вказані значення параметрів, в число яких входять параметри, вказані в переліку можливих параметрів функцій.

Переліки параметрів функцій, функціональних і конструктивних блоків залежать від їх типів і можуть бути представлені окремими інформаційними структурами.

Необхідно також враховувати, що в загальному випадку як функціональний блок може розглядатися як апаратний, так і програмний продукт без вказівки, яким саме типом продукту блок представлений. В ролі конструктивного блоку може бути представлено конкретний апаратний, програмний або апаратно-програмний засіб. Можливий вид запису в бібліотеці відображень (таблиця 3.1).

Створення бібліотеки відображень є одним з найбільш трудомістких і відповідальних завдань побудови систем проектування АЛП і вимагає детального опрацювання великої кількості схемних і структурних рішень. При ручному проектуванні застосування способів реалізації функцій ґрунтується на особистому досвіді розробника.

Перед побудовою бібліотеки відображень необхідно ретельно визначити наочну область, об'єм якої для невеликої системи структурного проектування

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

АЛП складає біля декількох сотень понять, що розбиваються на класи: об'єкт, визначеність, дія (в т.ч., відношення).

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ЗАГАЛЬНИЙ ПОРЯДОК ПРОЕКТУВАННЯ АЛП НА ОСНОВІ ФУНКЦІОНАЛЬНО-СТРУКТУРНОГО ПІДХОДУ

У даному розділі розглядаються загальні питання проектування структур АЛП з погляду концепції функціонально-структурного підходу та апарату еволюційного синтезу систем, що базується на ній [5]. На початку розділу приводяться основні положення цієї концепції. На її основі розбираються етапи проектування АЛП, відповідно до яких формується методика побудови узагальнених структурних схем АЛП. Ця методика ілюструється прикладом побудови АЛП для додавання чисел з плаваючою комою. У завершальній частині розділу розглядаються перетворення структур АЛП, що дають можливість варіювати апаратні витрати на їх реалізацію, а також можливі оцінки ефективності отримуваних структур.

#### 3.1 Формування функціональної структури АЛП

Після визначення концепції реалізації АЛП і побудови його дерева функцій (не обов'язково для всіх рівнів відразу) на відповідних етапах починається етап формування функціональної структури АЛП. На цьому етапі вибирається алгоритм реалізації потрібних функцій пристрою та здійснюється перехід від операторів (блоків) алгоритму до функціональних блоків АЛП. Причому, як наголошувалося вище, виконання цього етапу зазвичай супроводжується ітеративними поверненнями до етапу формування дерева функцій для виділення функцій нижчого рівня та відповідних йому операторів у вибраному алгоритмі з метою визначення способів реалізації операторів функціональними (а згодом і конструктивними) блоками.

					Ошибка! Источник ссылки не найден. ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

Відповідно до стратегії максимального розподілу функцій по функціональних блоках перехід від операторів алгоритму до функціональних блоків здійснюється за допомогою таких процедур:

1. По аналогії з правилами структурного програмування, алгоритм розділяється на ділянки або блоки, що мають один вхід і один вихід (при цьому послідовні блоки, що входять в крупнішу лінійну ділянку, спершу не виділяються, тобто, розбиття спочатку проводиться на найкрупніші ділянки).

2. Для кожної виділеної ділянки алгоритму (УчА) визначаються множина *In* вхідних і множина *Ot* вихідних інформаційних об'єктів (ІФО). При цьому в множині *Ot* вихідних ІФО включаються всі змінні, значення яких обчислюються операторами даної ділянки алгоритму, а в множину *In* вхідних ІФО включають решту всіх змінних. Для ділянок, що містять умовні оператори (тобто для ділянок алгоритму типу if-then, if-then-else), до вхідних ІФО відносять також і змінні, що визначають значення умов операторів if.

3. Кожній виділеній ділянці алгоритму ставиться у відповідність узагальнений функціональний блок (ОФБ) створеної функціональної структури пристрою. Кількість входів і виходів таких блоків береться рівною кількості інформаційних відповідних об'єктів, що відповідають УчА множини *In* і *Ot*.

4. Проводиться аналіз наявності в бібліотеці відображень типових функціональних блоків, що забезпечують реалізацію функцій, що виконуються операторами тих ділянок алгоритмів, яким відповідають введені узагальнені функціональні блоки.

Таблиця 3.1 – Фрагмент можливого варіанту бібліотеки (таблиць) відображень

Параметри

Оточення дії

Функція	дія	актанта	актор		засоб	середови	функціональ	спосіб	конструктив

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

			а		у	ща	ний блок	реалізації	ний блок
						дія			
Підсумовування двійкових кодів	Час Вірогідність помилок	Розрядність					Суматор	Паралельний Послідовний Табличний	Ім'я файлу конструктивних реалізацій суматора
Передача двійкового коду	Час Відстань	Розрядність		Вид зв'язку		Перешкоди	Блок вентилів Мультиплексор Модем		Ім'я файлу конструктивних блоків
Розпізнавання виклику (у мережі ЕОМ)	Час, Вірогідність пропуску помилок ого виклику	Структура коди виклику Частота передачі коди		Тип лінії передачі і Хвильовий опір		Амплітуда перешкоди Величини зовнішніх полів	Селектор Мережевий контроллер	Виявлення посліжки Прийом посліжки Порівняння з еталоном Реакція "та" Реакція "ні"	>> >>

Якщо такі типові блоки є в бібліотеці, то ОФБ позначаються ім'ям типового блоку з вказівкою значень його структурних (не фізичних) параметрів, наприклад кількості та розрядності, входів і виходів блоку. Ці значення визначаються характеристиками інформаційних об'єктів блоку.

Якщо в бібліотеці немає типового блоку, що забезпечує виконання необхідних функцій, то даний ОФБ не може бути реалізований безпосередньо. Тому та ділянка алгоритму, якому відповідає ОФБ, повинна бути піддана подальшому розділенню, що проводиться згідно пункту 1.



Виконання даних процедур завершується тоді, коли для всіх узагальнених функціональних блоків будуть знайдені відповідні їм типові функціональні блоки. Тому видно, що процедура переходу до функціональної структури в цілому носить ітеративний характер.

Результатом виконання перерахованих процедур є узагальнена структурна схема (ОСС), функціональні блоки якої відповідають ділянкам початкового алгоритму та зв'язані між собою по входах і виходах відповідно до використовуваних і формованих інформаційних об'єктів. Така схема має вигляд, що максимально розпаралелює для початкового алгоритму.

Певні особливості має виконання даних процедур для випадку циклічних ділянок алгоритму. У різних варіантах організації циклу може бути сформоване різна кількість інформаційних об'єктів. Число їх може бути як постійним, так і змінним. У останньому випадку зазвичай відома максимальна кількість отримуваних ІФО. Як правило, цикл вводиться в алгоритм при орієнтації його на послідовний характер перетворень і може бути завжди представлений паралельними гілками (однотипними функціональними блоками), відповідними всім отримуваним в циклі вихідним інформаційним об'єктам. Звичайно, повне розпаралелювання циклу приводить до отримання громіздких структур, але на даному етапі це не повинно відразу приводити до виключення подібних варіантів з подальшого аналізу.

Розглянемо виконання процедур на прикладі формування функціональної структури АЛП для додавання чисел з плаваючою комою.

Хай спочатку алгоритм виконання цієї операції представлений в загальному вигляді (рисунок 3.1).

Відповідно до процедури пункту 1, приведеного вище, алгоритм розбивається на дві ділянки, перші з яких складаються з блоку 1, а другий – із решти всіх блоків і має тип if-then-else.

Далі, згідно пункту 2 визначаються вхідні та вихідні інформаційні об'єкти для виділених ділянок алгоритму. Хай операндами даної операції є два числа  $X$  і  $Y$ , представлені у формі з плаваючою комою, в якій кожне з чисел зображається

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

мантисою  $M$  і порядком  $P$ , тобто  $X = \langle M_x, P_x \rangle$ ,  $Y = \langle M_y, P_y \rangle$ . Тоді для першої ділянки алгоритму множина вхідних інформаційних об'єктів  $In_1$  складатиметься з порядків  $P_x$  і  $P_y$ , а множина  $Out_1$  вихідних ІФО цього УЧА включатиме різницю  $P_x - P_y$  та об'єкт, вказуючий на те, який з порядків  $P_x$  або  $P_y$  більше і тому є попереднім порядком результату. В ролі такого інформаційного об'єкту-показчика може виступати спеціальна ознака або просто знак різниці  $P_x - P_y$ .

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50



Змн.	Арк.	№ докум.	Підпис	Дата

Рисунок 3.1 – Алгоритм додавання чисел з плаваючою комою

Для другого УчА множина  $I_{n2}$  вхідних ІФО включає додатково до вихідних ІФО першого блоку самі мантиси  $M_x$  і  $M_y$  та порядки  $P_x$  і  $P_y$  операндів, а множина  $O_{n2}$  вихідних інформаційних об'єктів складається з мантиси  $M_z$ , та порядку  $P_z$ , числа - результату  $Z$  (суми) і ознак результату  $F_z$ . (До ознак зазвичай відносяться ознака переповнення  $OVF$ , ознака втрати значущості  $SLS$ , ознака зникнення порядку  $ODA$ , ознака нульового результату  $N$ .)

Після цього, відповідно до процедури пункту 3, приведеного вище, будується узагальнена структурна схема (функціональна структура) АЛП, в якій кожному УчА поставлений у відповідність узагальнений функціональний блок, що реалізує функції цього УчА. На цій схемі для кожного такого ОФБ входи та виходи позначаються відповідними інформаційними об'єктами.

Отримано структурну схему АЛП (рисунок 3.2), де  $P_z^0$  – попередній порядок результату (дорівнює більшому з порядків доданків).

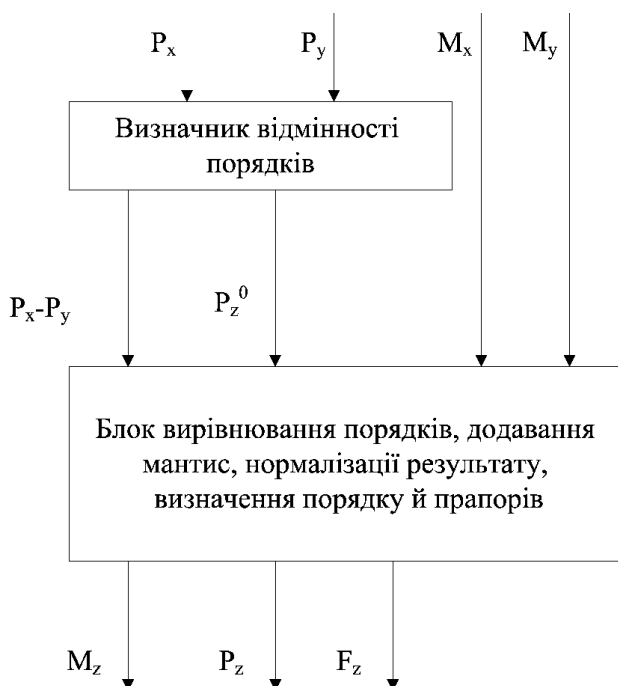


Рисунок 3.2 – Узагальнена структурна схема (функціональна структура) пристрою, що реалізує алгоритм додавання чисел з плаваючою комою при

розбитті його на дві ділянки

Виконання такої процедури (пункт 4) пов'язане з аналізом наявності в бібліотеці відображень типових блоків, відповідних узагальненим функціональним блокам побудованої схеми. Оскільки склад бібліотеки типових блоків поки не уточнений, то в даному випадку можна прийняти за типові блоки стандартні вузли ЕОМ.

В ході порівняння отриманих ОФБ з типовими блоками (ТБ) можна встановити, що перший ОФБ, що визначає різницю двох цілих чисел із знаками (порядків), реалізується на основі типового блоку – суматора (поки не уточнюється, як враховуються знаки та вибирається більше з чисел; таке уточнення буде зроблено в іншій частині). Проте другий ОФБ отриманої схеми не може бути реалізований одним типовим блоком хоч би тому, що в ньому послідовно виконуються різні функції. Звичайно, стосовно першого ОФБ також є певна умовність, оскільки і цей блок, окрім віднімання, здійснює ще й видачу більшого числа, тобто мультиплексування.

Тому для другого УчА необхідно знову провести всі розглянуті процедури, як для окремого алгоритму. При цьому початковий блок if даної ділянки слід виділити в самостійний блок (ділянку), так як інші правила структуризації не дозволяють провести подальше розбиття. Таке виділення блоку if викличе появу на структурній схемі самостійних паралельних блоків, відповідних обом гілкам ділянки, що розділяється. У свою чергу це вимагає включення в схему додаткового мультиплексора в кінці даної ділянки для вибору результату роботи одного з паралельних блоків схеми.

Після розбиття другого УчА початкового алгоритму буде отримано п'ять нових УчА, відповідних блокам 2...6 початкового алгоритму. Зіставлення узагальнених функціональних блоків приводить до отримання узагальненої структурної схеми (рисунок 3.3). На цьому рисунку прийняті такі позначення:  $G_i$  – показник числа, що має більший порядок;  $M_x'$  і  $M_y'$  – мантиси доданків після виконання процедури вирівнювання порядків;  $M_z'$  – ненормалізована мантиса

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

суми;  $B_i$  – значення логічної умови, що встановлюється за результатом порівняння модуля різниці порядків з числом розрядів мантиси;  $M_z^1, P_z^1, F_z^1$  і  $M_z^2, P_z^2, F_z^2$  – мантиси, порядки та ознаки результату, що отримуються в блоках правої та лівої гілок схеми. Крім того, шостому УчА блоку алгоритму (див. рисунок 3.3) поставлено у відповідність відразу три блоки ОФБ61, ОФБ62 і ОФБ63, об'єднаних пунктирно в загальний блок ОФБ1. У схемі також є додатковий блок мультиплексування ОФБ7 для видачі на вихід пристрою результату з відповідної частини схеми, оскільки в ній є дві паралельних гілки, які формують результат різними способами, про що наголошувалося вище.

Блок ОФБ1 в порівнянні з першим блоком (див. рисунок 3.2) має додатковий вихід, що вказує, який з порядків більше, точніше, якому доданку  $X$  або  $Y$  належить більший порядок. Інформація ця необхідна для вибору відповідної мантиси в блоці ОФБ3, що формує результат без виконання операції: вважаючи його рівним доданку з великим порядком. Інформаційний об'єкт на даному виході позначений через  $b_i$ ; і може бути як складовим номером з великим порядком, так і просто логічною змінною, що приймає, наприклад, значення 1, якщо більшим є порядок доданку  $X$ , та нуль – в протилежному випадку. Як  $G_i$  зручно використовувати знак різниці порядків.

На виході блоку ОФБ2 формується інформаційний об'єкт  $B_i$ , що є логічною змінною та визначає, яка з гілок алгоритму повинна бути реалізована. Ця змінна, що виробляється схемою порівняння, приймає, наприклад, одиничне значення, якщо модуль різниці  $P_x - P_y$  перевершує за величиною число розрядів мантиси та результат дорівнює доданку з великим порядком, та нулю – в протилежному випадку.

Проте змінна  $B_i$  поступає не на вхід блоку ОФБ3, а на вхід мультиплексора ОФБ7, в який також поступають результати з виходів блоків, що реалізують обидві гілки алгоритму.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

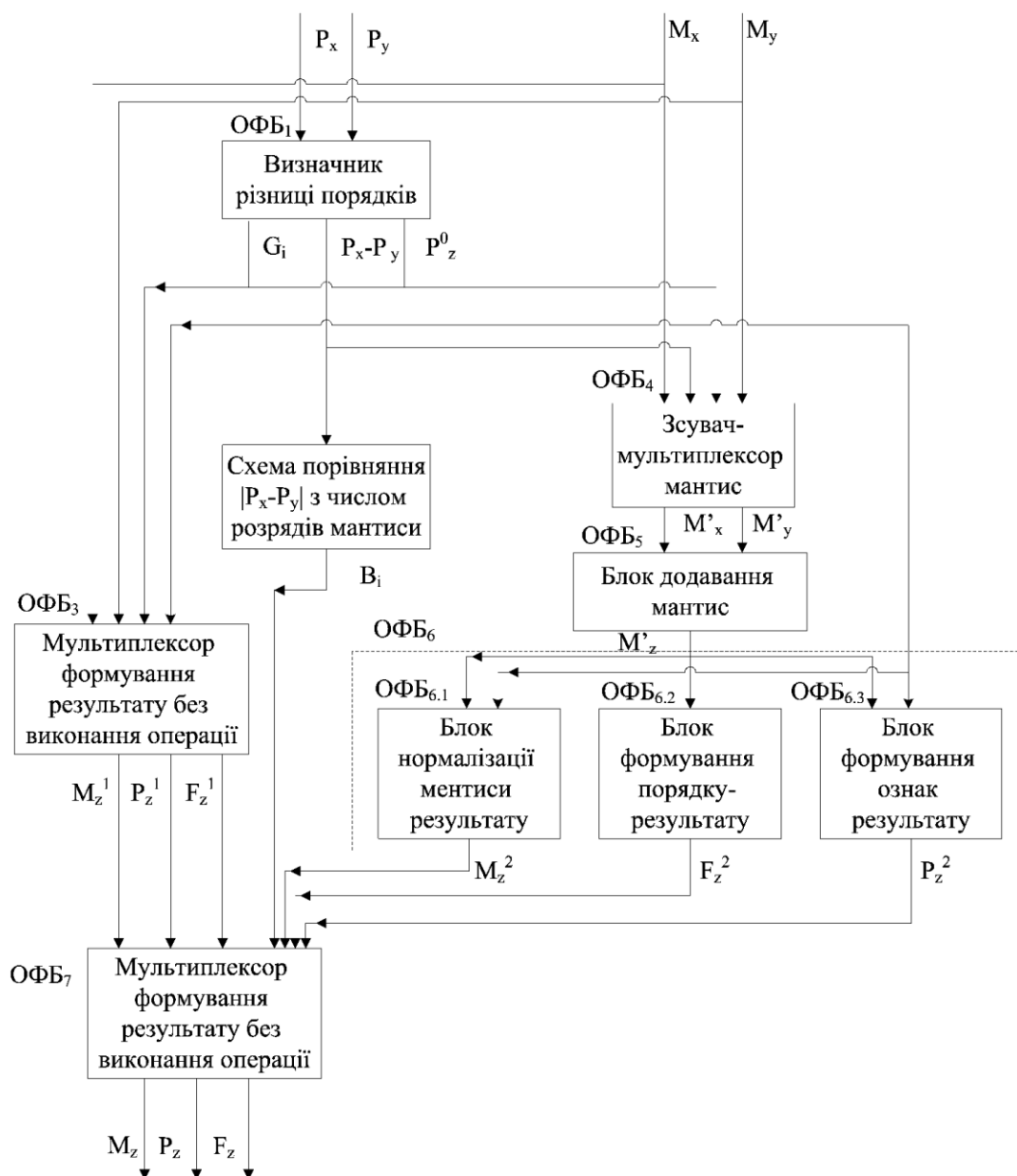


Рисунок 3.3 – Узагальнена структурна схема пристрою, що реалізує алгоритм додавання з плаваючою комою (рисунок 3.1), при розбитті алгоритму на шість ділянок (блоків)

У блоці ОФБ3, крім вибору мантиси доданку з більшим порядком, що здійснюється мультиплексором, проводиться також формування ознак результату, рівного цьому доданку.

На виходах блоку ОФБ4 формуються інформаційні об'єкти  $M'_x$  і  $M'_y$ , відповідні мантисам складових після вирівнювання порядків. При цьому мантиса числа з великим порядком проходить через блок без змін, а друга

мантиса зсувається вправо на число розрядів, що дорівнює різниці порядків. Таким чином в блоці ОФБ4 проводиться зсув та мультиплексування.

Блок ОФБ5 повинен виконувати додавання двох мантис з урахуванням їх знаків. За реалізованою функцією цей блок близький до блоку ОФБ1 і будується на базі суматора.

Блок ОФБ6 є розділеним на три функціональні блоки, оскільки дії (функції), перераховані у відповідному йому блоці 6 початкового алгоритму, є достатньо складними та звичайними типовими блоками не реалізуються (див. рисунок 3.3). Тому даний блок алгоритму повинен бути представлений в іншій формі, придатнішій для реалізації типовими блоками.

Як наголошувалося вище, завдання такого представлення відноситься до декомпозиції дерева функцій системи. Проте, по-перше, в даному випадку в блоці алгоритму задана сукупність трьох різних функцій, для виконання яких досить отриманих раніше інформаційних об'єктів. По-друге, не був прийнятий критерій оцінки системи, що розроблялася, а використовується стратегія максимального розпаралелювання функцій, що залишає можливості для оптимізації структури на подальших етапах. Тому блок 6 алгоритму представлений у вигляді трьох паралельно виконуваних функцій і йому зіставлені три окремих ОФБ: ОФБ6.1, ОФБ6.2 і ОФБ6.3 (Можлива організація деяких інформаційних зв'язків між цими блоками, наприклад, з ОФБ6.1 в ОФБ6.2 може поступати приріст порядку, визначуваний в процесі нормалізації мантиси, на яке слід змінити попередній порядок результату для набуття остаточного його значення).

У побудовану узагальнену структурну схему АЛП входять функціональні блоки, в основі кожного з яких лежить типовий блок (суматор, зсувач, мультиплексор). Але в їх склад входять також і додаткові елементи та вузли. Тому такі блоки можуть вважатися типовими лише в тому випадку, якщо їх проектування вже було виконане раніше і вони введені до складу бібліотеки типових блоків. Якщо ж прийняти за типові блоки тільки ті базові вузли ЕОМ (див. таблицю 2.2), то точної відповідності між ОФБ побудованої схеми АЛП і

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		



типовими блоками встановити не можна, оскільки функції цих ОФБ дещо складніші, ніж функції, що реалізуються типовими блоками. Тому для отримання схеми, що складається з типових блоків, необхідно провести декомпозицію функції, що реалізується блоками початкового алгоритму до простіших функцій, виконання яких забезпечується базовими вузлами.

Здійснення такого розбиття функцій розглянемо на прикладі блоку ОФБ5, оскільки проведення цього аналізу для всіх блоків є трудомістким. Крім того, блок ОФБ5, по-перше, близький по функції, що реалізується, до блоку ОФБ1, а по-друге, співпадає по виконуваному перетворенню з операцією додавання з фіксованою комою.

Блок 5 початкового алгоритму задає операцію додавання алгебри мантис. Декомпозиція цієї операції може здійснюватися декількома різними способами. Оскільки спеціальних критеріїв вибору способу декомпозиції тут не використовується, то за основу доцільно взяти алгоритм додавання чисел, представлених в прямому коді. Функції такого алгоритму (операції) можна розподілити по трьох його блоках (див. рисунок 3.3). Відповідна цьому алгоритму узагальнена структурна схема пристрою, перехід до якої проводиться за тими ж правилами (рисунок 3.4). Причому функціям блоків 5.1 і 5.3 алгоритму поставлені у відповідність окремі (по два на кожен) функціональні блоки (рисунок 3.5).

Початок

5.1  
Отримання  
зворотних кодів  
доданків

5.2  
Додавання кодів

5.3  
Отримання прямого  
коду суми.  
Визначення  
значення ознаки  
переповнення

Кінець

Рисунок 3.4 – Алгоритм додавання мантис

У цій схемі вхідними інформаційними об'єктами для блоків ОФБ5.11 і ОФБ5.1.2 є мантиси  $M_x$  і  $M_y$  (або просто числа з фіксованою комою) доданків, що складаються із знаку  $S$  і цифр  $D$  кожна, тобто,  $MX = \langle S_x, D_x \rangle$ ,  $M_y = \langle S_y, D_y \rangle$ . Вихідними ІФО даних блоків є цифри  $D_x'$  і  $D_y'$  зворотних кодів мантис, які разом із знаками утворюють собою вхідні ІФО блоку ОФБ5.2, який є суматором. Причому блоки ОФБ5.1.1 і ОФБ5.1.2 утворюють паралельні кола схеми

					Ошибка! Источник ссылки не найден. ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

відповідно до прийнятої стратегії максимального розподілу функцій алгоритму по блоках схеми. На виході блоку ОФБ5.2 формуються ІФО: знак  $S_z$ , та цифри  $D_z'$  зворотного коду  $M_z'$  мантиси суми, які перетворюються блоком ОФБ5.3.1, повністю аналогічним блоком ОФБ5.1.1 і ОФБ5.1.1 в прямий код мантиси суми  $M_z = \langle S_z, D_z \rangle$ .

Крім того, вихідними ІФО блоку ОФБ5.2 є сигнали перенесення  $C_0$  і  $C_1$  із знакового і старшого цифрового розрядів суматора, що поступають в блок ОФБ5.3.2 для встановлення значення ознаки переповнювання  $OVF$ , що є вихідним інформаційним об'єктом даного блоку. Слід зазначити, що значення ознаки переповнювання може визначатися інакше, наприклад, за значеннями основного  $S_z$ , та додаткового  $S_z^d$  знакових розрядів суми (див. рисунок 3.5), або по трьох знаках: доданків і результату.

Отримана функціональна структура (узагальнена структурна схема) пристрою містить типові блоки, що реалізують прості для структурного представлення АЛП мікрооперації: додавання, отримання зворотних і прямих кодів, логічну операцію «Виключе АБО» (для обчислення значення ознаки переповнювання  $OVF$ ).

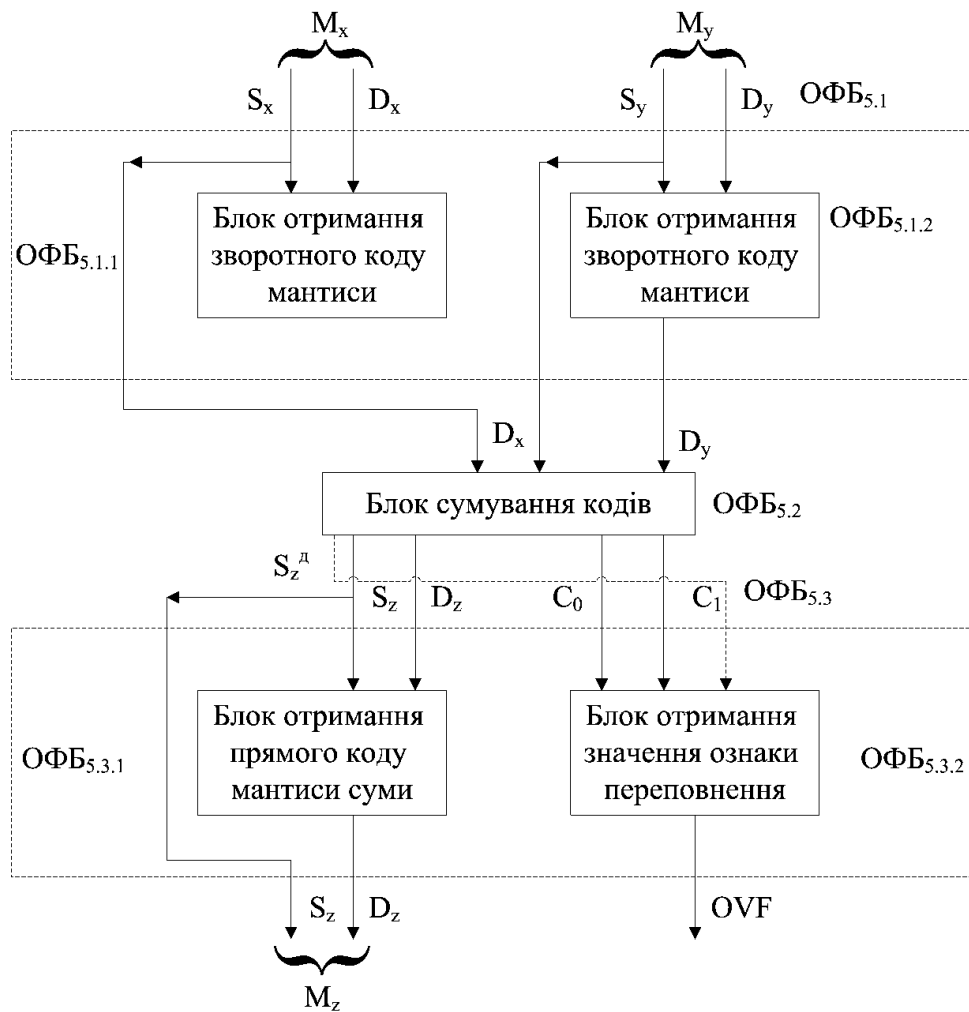


Рисунок 3.5 – Узагальнена структурна схема пристрою, що реалізує алгоритм додавання мантис

Заміна узагальнених функціональних блоків отриманої схеми на типові функціональні блоки дає структурну схему (рисунок 3.6). На цій схемі узагальнені функціональні блоки отримання зворотних і прямих кодів OFB5.1.1 OFB5.1.2 і OFB5.3.2 представлені блоком інверторів, що інвертує цифрові  $D$  розряди коду, та мультиплексора MUX, керованого знаковим  $S$  розрядом перетворюваного коду та  $S_m$  (або на вихід пристрою), що комутує на вхід суматора, прямі або інверсні значення цифр. Це пояснюється тим, що блок отримання зворотного коду зіставлено з блоком типу *if-then* алгоритму та передбачає два способи формування вихідного інформаційного об'єкту. Отже, відповідний йому структурний блок (як наголошувалося вище) повинен містити

мультиплексор, що вибирає один з двох формованих ІФО як вихідного. Втім, можлива й інша реалізація цього блоку безпосередньо із співвідношення для цифр зворотного коду (3.1):

$$D_i' = SRev(D_i) \vee Rev(S)D_i, \quad (3.1)$$

де  $D_i'$  -  $i$ -та цифра ОК;

$D_i$  -  $i$ -та цифра початкового коду;

$S$  – знак коду.

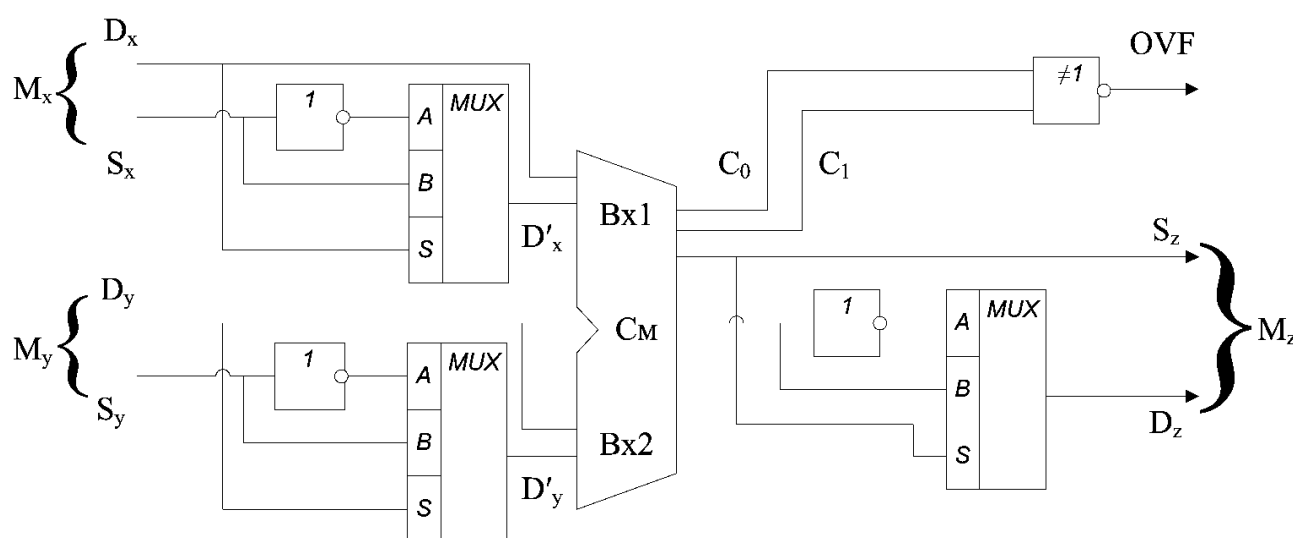


Рисунок 3.6 – Структурна схема пристрою, що реалізує додавання мантис

Аналогічним чином можуть бути отримані структурні схеми для всіх решти ділянок даного початкового алгоритму додавання чисел з плаваючою комою.

Результатом виконання етапу формування функціональної структури буде структурна схема пристрою, що складається з типових (комбінаційних) функціональних блоків, що виконує всі операції описані початковим алгоритмом, без керуючих сигналів та буферизації (запам'ятовування) проміжних результатів. Іншими словами, можна вважати, що побудована схема виконує задану операцію за один (достатньо тривалий) такт. Така схема

відповідає граничному розподілу виконуваних функцій за функціональними блоками, коли кожен з блоків виконує тільки одну функцію.

Таким чином, отримання даної схеми можна було б розглядати як завершення етапу формування функціональної структури АЛП. Проте процедура розробки її припускала побудову структури з максимальним розпаралелюванням, яка надалі може піддаватися перетворенням з метою зменшення апаратних витрат. Певні перетворення можуть бути і на етапі формування конструктивної структури (див. перелік етапів проектування). Якщо ці перетворення не зачіпають алгоритмів, узятих за основу при виконанні етапу формування функціональної структури, то її модифікації не вимагають повторення даного етапу. Якщо ж зміни проводяться в самому алгоритмі, то це може привести до необхідності сформувати нову функціональну структуру пристрою.

Розглянемо такі основні перетворення.

### 3.2 Модифікація алгоритмів і структур АЛП

Між перетвореннями алгоритмів і функціональних структур є певні аналогії, оскільки як перші, так і другі описують деяку послідовність функцій. Але між цими перетвореннями є також відмінності, тому доцільно розглянути їх окремо.

#### 3.4.1. Модифікація алгоритмів виконання операцій

Основні перетворення алгоритмів, еквівалентні у функціональному відношенні, які можуть використовуватися при проектуванні АЛП, зводяться до таких груп:

- зміна порядку проходження блоків алгоритму;
- заміна функціонального базису алгоритму;
- заміна частини або всього алгоритму.

До зміни порядку проходження блоків алгоритму відносяться:

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

а) перестановка місцями незалежних блоків (ділянок) алгоритму. Причому незалежність блоків розуміється в двох аспектах: у інформаційному сенсі, як відсутність в них загальних змінних, значення яких можуть змінюватися блоками, та в логічному сенсі, як відсутність умов (усередині або поза блоками), одні й ті ж значення яких по-різному впливають на виклик даних блоків на виконання. При перестановці місцями двох незалежних блоків алгоритму інформаційна та логічна незалежність повинна також розповсюджуватися на всі блоки, що потрапляють в область перестановки (між блоками, що переставляються);

б) розпаралелювання (перехід до одночасного виконання) незалежних ділянок або блоків алгоритму;

в) перехід до послідовного виконання одночасно виконуваних незалежних ділянок або блоків алгоритму.

До перетворень, пов'язаних із заміною функціонального базису алгоритму, відносяться:

а) об'єднання (композиція) функціональних блоків алгоритму, що приводить до появи нової функції (наприклад, послідовний рахунок – додавання, послідовність додавань – множення, обчислення результату – використання таблиці результатів);

б) поділ (декомпозиція) функціональних блоків алгоритму (наприклад, множення – серія додавань);

в) використання інших типів операцій (наприклад, віднімання – додавання із зворотним колом, зсув вліво – додавання числа з самим собою).

Прикладами заміни алгоритму або частини його, що відносяться до третьої групи, можуть служити переходи до використання різних методів або формул для вирішення одного і того ж завдання. Так, підрахунок площі  $S$  трикутника можна робити за формулою  $S = b \cdot h / 2$  або замість цього за формулою  $S = 1/2 \cdot a \cdot b \cdot \sin \gamma$ , суму чисел відрізка натурального ряду від  $n_1$  до  $n_2$  можна отримати послідовністю додавань або за формулою  $(n_2 \cdot (n_2 + 1) - n_1 \cdot (n_1 - 1)) / 2$ . Але такі заміни здійснюються за рахунок використання деякого

									Арк.
									63
Змн.	Арк.	№ докум.	Підпис	Дата	Ошибка! Источник ссылки не найден. ПЗ				

прихованого повністю або частково в алгоритмі взаємозв'язку між об'єктами та операцій перетворення. І в цьому сенсі заміни можуть відноситися не стільки до самого алгоритму, скільки до вищого рівня знань про перетворювані алгоритмом об'єкти, вирішувані завдання. Так, в приведених прикладах це знання про співвідношення елементів трикутника та натурального ряду чисел.

Перетворення, віднесені до різних груп можуть бути близькими один одному, наприклад, такими є "б" з першої групи та "а" з другої, перетворення "в" з другої групи до перетворень третьої групи та т.д.

#### 3.4.2. Модифікація структури АЛП

Після виконання етапу формування функціональної структури АЛП відповідно до прийнятої стратегії отримують максимальне розпаралелювання функцій. Перетворення структури пристрою в загальному випадку може переслідувати різні цілі, пов'язані з поліпшенням тих або інших їх характеристик. Найчастіше як такі розглядаються швидкодія та апаратні витрати.

##### Підвищення швидкодії

У структурі з максимальним розпаралелюванням функцій підвищення швидкодії можливе тільки за рахунок використання більш швидкодіючих елементів або зменшення довжини кіл перетворення (глибини схеми) пристрою, що дає зменшення тривалості такту та підвищення робочої частоти.

Перший із способів не пов'язаний з модифікацією структури. А зменшення глибини схеми безпосередньо припускає структурні зміни та може бути виконане в структурах з максимальним розпаралелюванням функцій двома шляхами:

- зменшенням кількості блоків та їх спрощенням, головним чином, за рахунок виключення складних операцій з числа операцій, що реалізуються пристроєм;
- максимально можливим зменшенням кількості комутуючих вузлів (блоків вентилів, мультиплексорів) в колах пристрою.



Виключення складних операцій в принципі знижує функціональні можливості пристрою, проте цей шлях має свої переваги, що підтверджується життєздатністю EOM з RISC архітектурою (EOM з скороченим набором команд). Структура цих EOM орієнтована на виконання коротких і простих операцій, а довгі – реалізуються програмним шляхом.

Скорочення кількості комутуючих вузлів в структурі з максимальним розпаралелюванням функцій можливо в тих випадках, коли в алгоритмах операцій є розгалуження. В цьому випадку кожній з гілок може бути поставлений у відповідність свій блок перетворень, а комутація проводитиметься тільки на виході пристрою за допомогою вибору потрібної гілки.

Нарешті, зменшення глибини схеми можливе за рахунок використання блоків з меншою глибиною кіл. Наприклад, як суматор може використовуватися суматор з паралельними перенесеннями, що має найбільш короткі, за кількістю елементів, схеми формування сигналів перенесення.

#### Зменшення апаратних витрат

Зменшення апаратних витрат зазвичай розуміється як зменшення кількості блоків, що входять в пристрій, або їх спрощення. Якщо не допускати дроблення блоків, наявних в перетворюваній структурі, то здійснення модифікацій можливе лише за рахунок таких процедур:

1. Поєднання виконання однакових перетворень, відповідних різним блокам алгоритму, в одному та тому ж типовому функціональному блоці (наприклад, для розглянутого вище пристрою додавання мантис операції перетворення доданків в зворотний код і суми в прямий код можна покласти на один і той же блок).

2. Перенесення виконання деякого простішого перетворення  $P_i$  в блок, призначений для виконання складнішого перетворення  $P_j$ , що реалізується в пристрої в тому випадку, якщо перетворення  $P_i$ , можна розглядати як окремий випадок перетворення  $P_j$  (наприклад, додавання одиниці можна проводити не в лічильнику, а в суматорі, якщо такий вже є в пристрої).

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Зменшення кількості входів багатовходових типових блоків, що реалізують деяке перетворення перетворення  $P_i^n$  від  $n$  аргументів), за рахунок використання блоків, що реалізують те ж перетворення  $P_j^k$  від меншої кількості аргументів,  $k < n$ . Наприклад, заміна трьохвходового суматора двовходовим, перехід від паралельної обробки до паралельно-послідовної та послідовної порозрядної обробки.

Розгляд цих способів дає можливість відзначити таке.

Поєднання виконання декількох однакових перетворень в одному типовому блоці, як і скорочення кількості входів блоку, в більшій частині випадків вимагає переходу до багатотактної реалізації всього перетворення, буферизації проміжних результатів, і, відповідно, мультиплексування входів і/або виходів блоків. Тому використання суміщеного виконання перетворень виправдане з погляду зниження складності пристрою (точніше, спрощення його блокового складу, оскільки поява нових зв'язків також може вимагати вирішення додаткових завдань проектування, особливо в технології ВІС), якщо вводяться буфер і мультиплексори блоку, що простіше виключається. З цієї точки зору доцільне поєднання блоків, починаючи тільки з деякої граничної межі їх складності. Так, поєднання блоків вентилів, мультиплексорів, інверторів і т.д може бути недоцільним, якщо буферизація та мультиплексування (шинна організація) з певних причин не передбачені в пристрої.

Поєднання однакових перетворень в одному типовому блоці має свої особливості залежно від взаємного розташування блоків в пристрої, що реалізують ці перетворення. Так, в деяких випадках можна не використовувати буферизацію.

До основних варіантів взаємного розташування блоків структурної схеми пристрою, що виконують однакові перетворення, можна віднести:

- 1) розташування блоків в паралельних колах схеми;
- 2) розташування блоків в послідовному ланцюзі (колах) схеми;
- 3) змішаний варіант: розташування блоків і в паралельних і в послідовних

колах.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

Причому при розташуванні в паралельних колах можливі випадки такого використання результатів:

- з  $N$  результатів перетворень, що виконуються паралельно над  $N$  різними вхідними ІФО, використовується тільки один, вибраний за значенням деякої умови, що позначатиметься через  $N \rightarrow 1$ ;
- використовуються всі  $N$  результатів однакових перетворень над  $N$  різними вхідними ІФО, що буде позначене як  $N \rightarrow N$ .

Наведено відповідні цим випадкам модифікації структури пристроїв, що зменшують кількість блоків перетворення (рисунок 3.7, а, б), де  $M$  – буфер для зберігання результату (регістр, тригер),  $DMX$  – демультіплексор.

Для кола  $N \rightarrow 1$  слід також розрізняти ситуації, в яких вибір залежить від значень вхідних ІФО або від результатів їх перетворення. Модифікація структури (див. рисунок 3.7, а), відповідає першому випадку. У другому випадку варіант принципово не відрізняється (див. рисунок 3.7, б). Видно, що кількість буферів, що вводиться при такій модифікації, повинна бути не менше  $N-1$ , а окрім вхідного мультиплексора, потрібний ще і демультіплексор, що забезпечує занесення чергового результату у відповідний буфер (див. рисунок 3.7, б).

Можливий і проміжний варіант, коли використовуються  $n$  результатів з  $N$  перетворень. Цей випадок може бути представлений як поєднання перших двох.

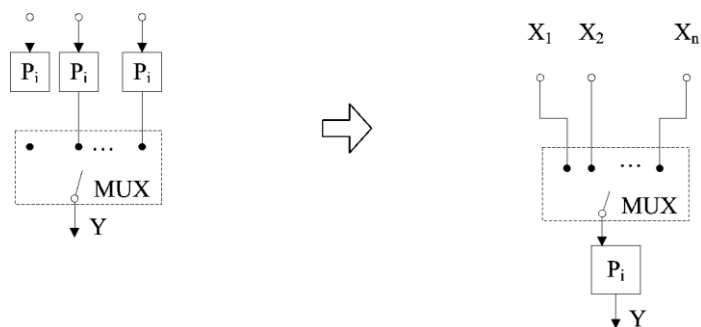
При розташуванні однакових блоків в послідовному ланцюзі їх поєднання також вимагає використання буфера, мультиплексора та демультіплексора, як і у попередньому випадку (див. рисунок 3.7, б). Проте буфер може бути поставлений в будь-якій частині кола, розташованого між суміщеними блоками. Відповідна цьому випадку модифікована структура (див. рисунок 3.7, в) з можливими різними розташуваннями буфера показані на рисунку пунктирними лініями: буфер може займати будь-яке одне з вказаних місць.

Другий спосіб модифікації структур пристроїв полягає в перенесенні виконання простих перетворень в наявні в пристрої блоки, орієнтовані на виконання перетворень, окремим випадком яких є переносимі прості

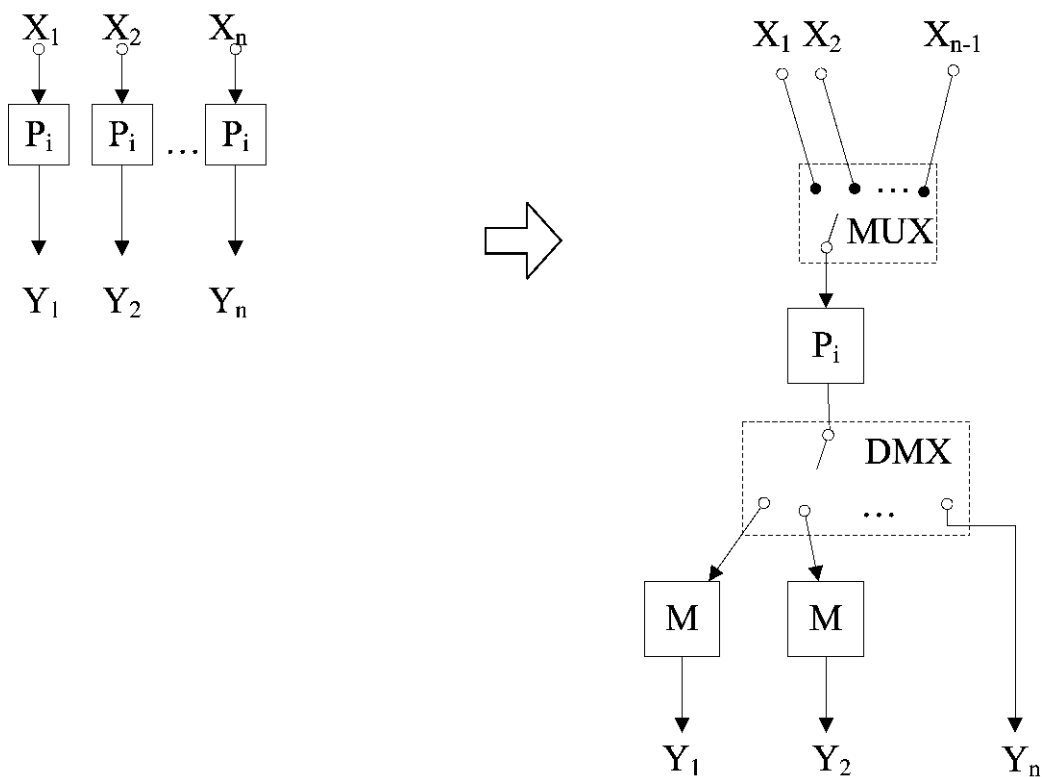
					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

перетворення. За отриманими варіантами модифікації структури цей спосіб повністю аналогічний розглянутому вище поєднанню однакових перетворень.

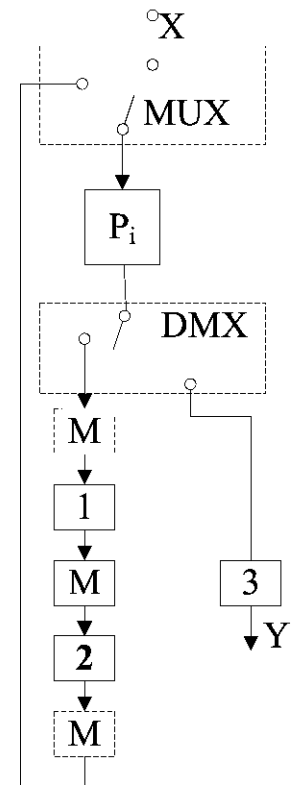
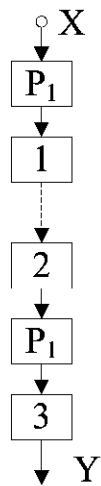
Третій з вказаних способів спрощення структури: зменшення кількості входів типових блоків за рахунок реалізації перетворень від великого числа аргументів однойменними перетвореннями від меншого числа аргументів (для перетворень, що допускають таке представлення, наприклад, для підсумовування), відповідає заміні одного багатовходового блоку ланцюжком блоків або деревом блоків з меншим числом входів.



a)



б)



в)

Рисунок 3.7 – Перетворення різних типів структур за рахунок поєднання виконання однакових функцій в одному блоці

Перехід до ланцюжка блоків (рисунок 3.8, а), де  $P^N-N$  – місцеве перетворення,  $P^2$  – однойменне двомісне перетворення. Такий перехід відповідає переходу від перетворення  $Y = P^N(X_1, X_2, X_N)$  до рекурсивної підстановки виду  $Y_i = P^2(X_i, Y_{i-1})$ , де  $i=2 \dots N$ ,  $Y_1 = X$ ,  $Y_n = Y$ .

До отриманого послідовного ланцюжка однакових блоків  $P^2$  може бути застосований перший з розглянутих способів перетворення структури пристрою: поєднання однакових операцій, що дає схему (див. рисунок 3.8, а).

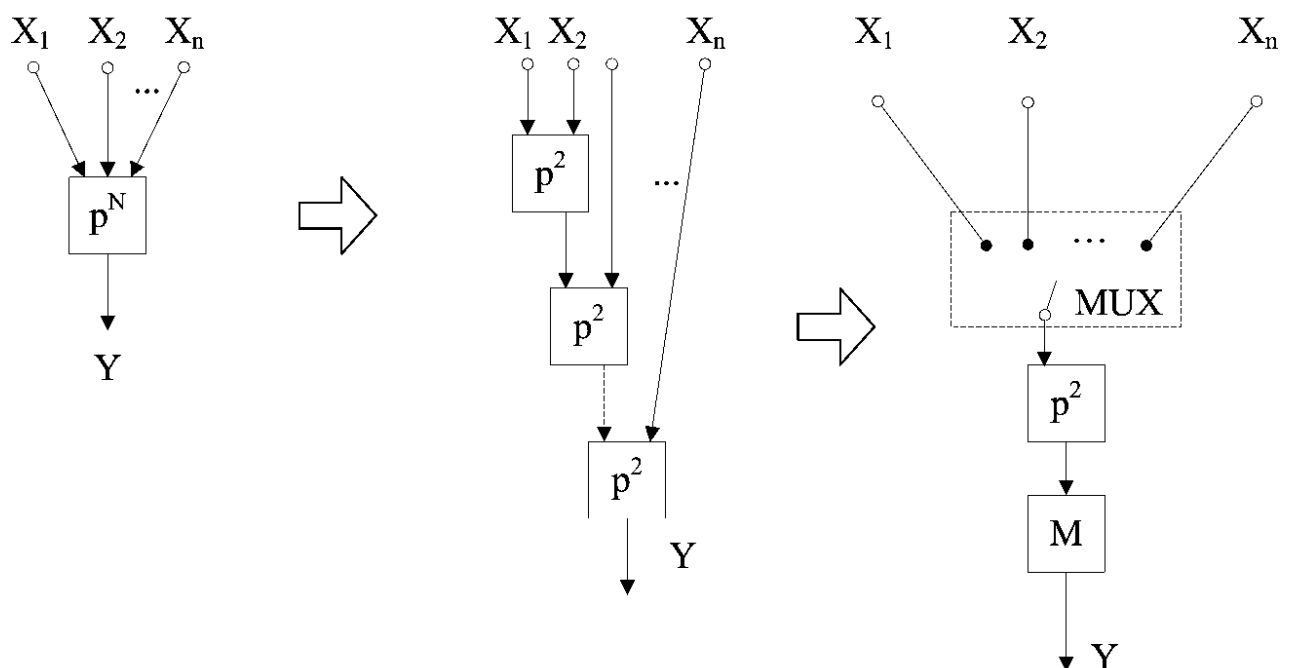
Заміна  $N$ -входового перетворювача на дерево блоків зберігає поярусну паралельність структури, яка виявляється більш швидкодіючою, в порівнянні з попереднім варіантом (рисунок 3.8 б).

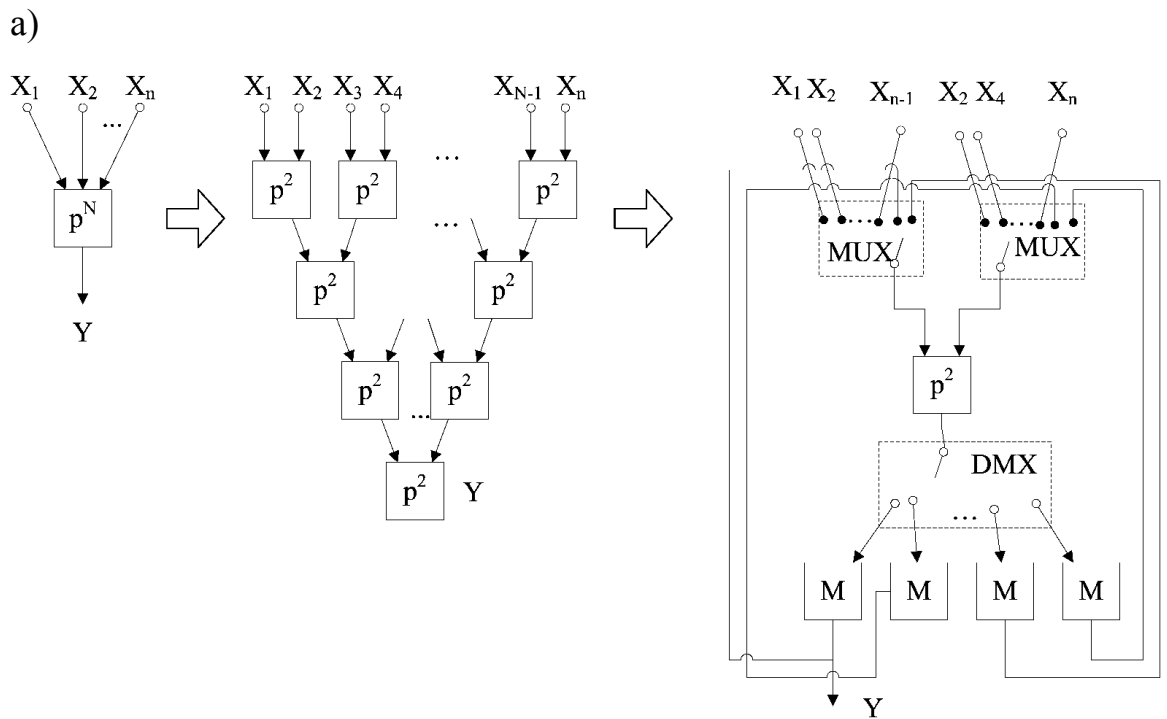
Кількість використовуваних при цьому блоків таке ж, як і в послідовному ланцюжку. Але істотно спростити отриману структуру не вдається, оскільки на першому ярусі дерева доводиться буферизувати  $N/n$  інформаційних об'єктів (де

$n$  – кількість входів блоків перетворення). Результат такого поєднання показаний в правій частині рисунка 3.8 б, де зроблені певні спрощення: оскільки частина буферів використовується багато разів (у декількох ярусах), то мультиплексори та демультимплексор повинні бути б мати більше положень: по декілька для багато разів використовуваних буферів.

При всіх розглянутих перетвореннях структури пристрою передбачалася наявність на входах блоків перетворення вхідних ІФО протягом всього часу перетворення. Таке допущення викликане обмеженням завдання проектування розробкою власне АЛП (або будь-якого окремого блоку, пристрою та т. д.), рішенням цієї задачі окремо від побудови інших пристроїв процесора. Це, по суті, еквівалентно виділенню, за правилами формування функціональної структури, ділянки загального алгоритму роботи процесора, відповідного виконанню операцій в АЛП.

Приведені варіанти перетворення структур пристроїв дають можливість здійснювати перехід від структури з максимальним розпаралелюванням функцій до структури з мінімальним складом блоків. При цьому виникають проблеми вибору та оцінки різних структур.





б)

Рисунок 3.8 – Перетворення структури, що дають зменшення кількості входів типових блоків за рахунок реалізації їх функцій ланцюжком (а) і деревом (б) типових блоків з меншою кількістю входів

### 3.3 Оцінка структур АЛП

До основних технічних характеристик пристроїв обробки інформації, що виконуються на сучасній технологічній базі відносять займану на кристалі площу, швидкодію, розсіювану потужність. Крім того, часто враховуються такі особливості схем, як тестопридатність, іноді, топологія зв'язків, що характеризує складність проектування та виготовлення схеми.

У загальному випадку для оцінки якого-небудь об'єкту може використовуватися деяка математична форма: критерій, що враховує різні характеристики оцінюваного об'єкту. Методи побудови та застосування таких критеріїв розглянуті в різних роботах, зокрема в [16]. Один з прикладів такого

критерію для АЛП враховує кількість елементів, вузлів, зв'язків, керуючих сигналів та час виконання операцій в АЛП.

Проте безпосередній зв'язок між особливостями та характеристиками власне структурної організації пристрою та характеристиками його технічної реалізації, таких як займана на кристалі площа, тестопридатність, прослідкувати досить складно. Розсіювану потужність для однакових за складом структур можна вважати приблизно однаковою.

Тому основна увага при порівнянні різних варіантів структур, що отримуються на етапі формування функціональної структури, буде приділено їх часовим оцінкам, які визначаються глибиною (кількістю послідовних елементів) отримуваних кіл перетворення. З цієї точки зору такт роботи пристрою, що виконує перетворення за декілька тактів і має вихідний буферний регістр для збереження проміжних результатів, можна розглядати як сукупність таких фаз:

- налаштування багатофункціональних блоків перетворення на виконання заданих функцій і налаштування мультиплексорів вхідних кіл перетворювачів;
- проходження сигналів через перетворювачі та виконання перетворень над ними;
- проходження сигналів через вихідні комутатори (демультиплексори) і стробування їх з метою «вирівнювання» в часі;
- буферизація результату.

Приблизне розташування цих фаз по робочому такту пристрою показано на рисунку 3.9, де  $t_m$  – тривалість такту,  $t_n$  – тривалість налаштування перетворювачів і мультиплексорів на їх входах,  $t_{np}$  – тривалість проходження сигналів через різні блоки або кола блоків перетворення,  $t_y$  – тривалість затримки сигналів на вихідних комутаторах і вентилях управління та  $t_\delta$  – тривалість буферизації результату перетворення. Видно, що дії, безпосередньо зв'язані з функціональними перетвореннями, становлять частину такту. Дії з комутації, настройки та управління, в загальному випадку, можуть бути пов'язані з особливостями алгоритмів, що реалізуються, та бути присутнім

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		



навіть в структурах з максимальним розпаралелюванням. Але більшою мірою ці дії пов'язані з мультиплексуванням і налаштуванням блоків перетворення, що суміщають виконання декількох однакових або різних функцій. Тому такі дії не можуть вважатися ефективними з погляду виконання перетворень, як не може вважатися ефективною та сама буферизація. Отже, можна визначити коефіцієнт  $n_p$ , корисного використання робочого такту як  $n_p = t_{np} / t_m$ . Тоді цей коефіцієнт буде тим вищий, чим складнішим є перетворення, чим менше мультиплексорів і демультимплексорів є в структурі та чим більшою є швидкодія буферів.

Типове «довге» перетворення – додавання при великій глибині кола перенесення дасть високий  $n_p$ , але ціною цьому буде нижча швидкодія пристрою, особливо при виконанні логічних операцій.

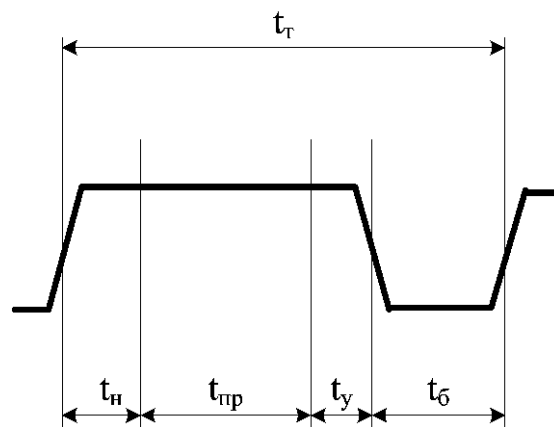


Рисунок 3.9 – Робочий такт пристрою з вихідним буфером

Можна вважати, що з двох структур однакового складу кращою є та, яка має вищий коефіцієнт  $n_p$ . Проте цей коефіцієнт не повністю визначає вибір структури. Адже якщо частота використання кола перетворення найбільшої глибини невисока або коло використовується переважно для виконання простіших дій (як окремих випадків такого перетворення), то може виявитися доцільним скоротити або вилучити це коло, не зважаючи на зниження коефіцієнта  $n_p$ .

## Дослідження поведінкової моделі арифметичного пристрою

Арифметичний пристрій, який представлений блоком В\_ARU, є центральною частиною практично будь-якого процесора і призначений для виконання арифметичних і логічних операцій з операндами, заданими в поточній команді. Перелік операцій і операндів визначений в системі команд процесора.

Перелік операцій В\_ARU визначає продуктивність процесора, з одного боку і складність В\_ARU і дешифратора команд, з іншого боку. Мінімальний перелік операцій містить операції, що забезпечують алгоритмічну повноту системи команд. Це, як мінімум, три логічні операції, наприклад, І, НЕ, встановлення в 0, арифметичні операції додавання, віднімання, а також операції зсуву на один розряд управо. Решта арифметичних і логічних функцій обчислюється шляхом виконання послідовності команд.

Для збільшення продуктивності процесора і скорочення довжини програм в перелік команд вводяться операції обчислення функцій, які часто зустрічаються в алгоритмах. Тому операції множення і ділення входять в систему команд більшості процесорів. Вибором переліку операцій є пошук компромісу між зручністю системи команд, об'ємом пам'яті програм, швидкодією процесора і складністю В\_ARU.

Операнди в В\_ARU можуть поступати з регістрової пам'яті В\_RRAM, оперативної пам'яті RAM або із спеціальних регістрів, наприклад, з регістра-акумулятора. Ці ж блоки і регістри є приймачами результату В\_ARU. Ознака перенесення С бере участь в операціях як особливий операнд.

Джерела операндів визначаються видами адресації системи команд. При використанні декількох видів адресації входи В\_ARU повинні містити мультиплексори операндів, які підключають ті або інші джерела операндів. Такі

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

мультиплексори вносять додаткові апаратні витрати і додаткову затримку до загальної тривалості виконання команд.

Для проведення проектування 16-розрядного В\_ARU, що виконує набір операцій (таблиця 3.2) на базі трьохкнального блоку В\_RRAM з вісьма регістрами. До складу В\_ARU входить блок В\_ALU, що виконує операції додавання і віднімання з перенесенням. Операції додавання і віднімання утворюються шляхом подачі 0, 1 або флага перенесення С на вхід С0 блоку В\_ALU (див. таблицю 3.2).

Для зсуву вправо необхідно підключити зсувач на один розряд. При цьому для виконання логічного, арифметичного зсуву або зсуву з перенесенням в лівий розряд результату вставляється 0, флаг С або флаг знаку N. Операція множення виконується послідовно-паралельно в блоці множення.

Таблиця 3.2 – Кодування операцій В\_ALU

Операція	Мнемоні-ка	Код операції АСОР	Код F управління перенесення В_ALU	Біт перенесення С або біт S15
Додавання	ADD	0000	00	0
Додавання з перенесенням	ADDC	0001	00	С
Додавання з інкрементом	ADDINC	0010	00	1
Віднімання з декрементом	SUBDEC	0011	01	0
Віднімання з перенесенням	SUBC	0100	01	С
Віднімання	SUB	0101	01	1
Зсув управо арифметичний	SRA	0110	-	N
Множення	MUL	0111	-	-

Оскільки блок В\_RRAM – трьохадресний, то за однією командою можна вибирати 2 джерела операндів і 1 приймач – результат. Оскільки використано 8 комбінацій кодів операції, то в систему операцій В\_ARU можна ще додати й інші операції.

Операція множення має 2 операнди і 2 шістнадцятирозрядних результати (частини множення). Оскільки задана одна адреса результату, то доцільно 2 слова результату розташовувати в двох сусідніх регістрах, адреси яких відрізняються на 1. Для цього на адресному вході AQ блоку В\_RRAM слід вставити схему інкремента.

Всі операції повинні видавати ознаки результату – біт перенесення, Z – ознаку нульового результату і біт знаку N, які повинні запам'ятовуватися в регістрі стану. При виконанні виклику підпрограм в області стека повинно зберігатися стан регістра стану і адреса повернення з підпрограми, а за командою повернення з підпрограми ця адреса повертається в В\_CCM. У RISC - процесорах при виклику підпрограм адреса повернення зберігається в одному з регістрів блоку В\_RRAM, наприклад, в останньому. Оскільки в даному процесорі адресний простір обмежений 13 розрядами, то доцільно молодші розряди регістра 7 відвести під адресу повернення, тобто біти C, Z і N зберігатимуться, відповідно, в 15, 14 і 13 розрядах цього регістра.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.4 Дослідження структурної моделі арифметичного пристрою

Для реалізації всіх функцій арифметичного пристрою В\_ARU необхідно сформулювати його структуру (рисунок 3.10).

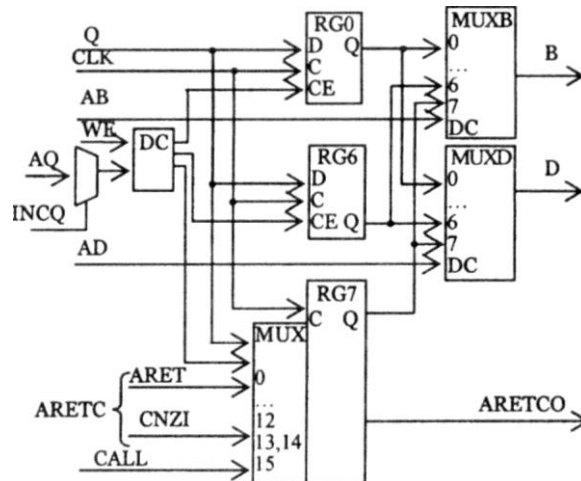


Рисунок 3.10 – Структура модернізованого блоку В\_RRAM

В структуру блоку В\_ARU окрім блоку В\_RRAM входять блоки MPU і В\_ALU (рисунок 3.11). Мультиплексор MUXQ призначений для подачі на вихід D0 блоку і на вхід запису Q вхідного даного, або результату В\_ALU, або результату MPU, або даних обчислень D, або його ж, але зсунутого вправо за командами SRL, SRC або SRA. Мультиплексор MUXC вибирає флаг перенесення C або 0, або 1, щоб подати на вхід перенесення В\_ALU для виконання команд додавання і віднімання.

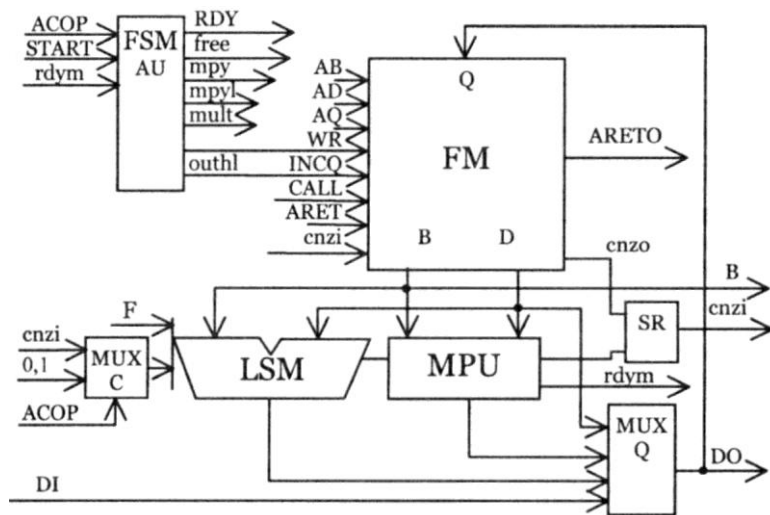


Рисунок 3.11 – Структура блоку B\_ARU

За допомогою регістра стану SR записуємо в кінці виконання команд ознаку перенесення C, ознаку негативного результату N і ознаку нульового результату Z, причому, якщо це команда множення, то стан записуємо з блоку множення, якщо виконується команда повернення з підпрограми RET – то стан, що зберігається в 7-му регістрі B\_RRAM, а інакше – з блоку B\_ALU.

На вихід B подамо дані, прочитані з B\_RRAM за адресою AB. Їх використаємо як адресу при індексній адресації зовнішньої пам'яті.

Керуючий автомат FSM\_B\_ARU має 3 стани: free – B\_ARU вільний, mpy – відбувається операція множення, mpyl – операція множення завершена. На основі них та вхідних сигналів ACOP, START rdym формуємо необхідні керуючі сигнали (рисунок 3.12).

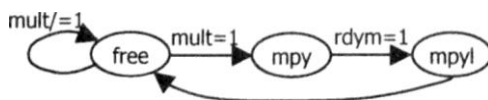


Рисунок 3.12 – Діаграма станів ESM\_B\_ARU

## 4 ОХОРОНА ПРАЦІ

Поняття “охорона праці” визначено статтею 1 Закону України “Про охорону праці” [11]. Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів і засобів, спрямованих на збереження здоров’я і працездатності людини в процесі праці [14].

Основною ціллю охорони праці є створення на кожному робочому місці безпечних умов праці, експлуатації обладнання, зменшення або повна нейтралізація дії шкідливих і небезпечних виробничих факторів на організм людини і зниження виробничого травматизму та професійних захворювань.

### 4.1 Аналіз санітарно-гігієнічних умов праці

У даному розділі розглядаються питання охорони [15] праці при розробці арифметичного пристрою RISC-процесора.

У приміщення працює до восьми чоловік. Види виконуючих робіт – вивчення матеріалів у відповідності з розвитком техніки, розробка, налагодження апаратного забезпечення, симуляція моделей на комп’ютерах. Для цих цілей у приміщення використовується 8 ЕОМ класу AMD Athlon XP 1800+ (1533 MHz), а також лазерний монохромний принтер формату А4 марки HP LaserJet 1020. На персональних комп’ютерах (ПК) встановлені монітори марки LG Studioworks 575C (розширення 1024x768).

Робочі місця розміщені вздовж периметра (рисунок 4.1). По три місця розміщено вздовж довших стін.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

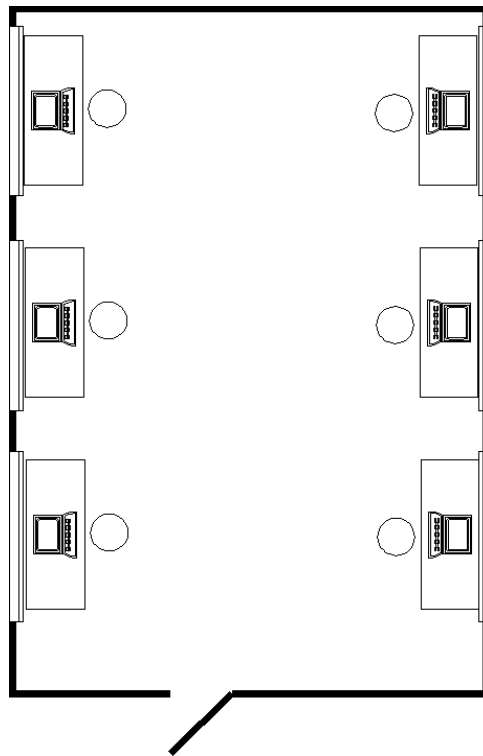


Рисунок 4.1 – План приміщення

Санітарно-гігієнічні умови праці розраховуються згідно з розмірами приміщення (таблиця 4.1).

Таблиця 4.1 – Розміри приміщення

Позначення	Визначення	Значення
$l$	Довжина	8 м
$d$	Ширина	6 м
$h$	Висота	4 м
$S_0$	Площа	$48 \text{ м}^2$
$V_0$	Об'єм	$192 \text{ м}^3$

Згідно СН-245-71, на одного працюючого об'єм приміщення повинен складати не менше  $19,5 \text{ м}^3$ , площа – не менше  $6 \text{ м}^2$  [7]. Число працюючих у приміщенні  $N_p=8$ . Таким чином, на одне робоче місце приходить площа  $S=48/8=6 \text{ (м}^2\text{)}$  і об'єм  $V = 192/8=24 \text{ (м}^3\text{)}$ . Ці значення відповідають вимогам.



При розрахунку санітарно-гігієнічних умов праці дотримуються норм приміщення (таблиця 4.2).

Таблиця 4.2 – Норми приміщення

Параметр	Значення
Ширина основних проходів	$\geq 1200$ мм
Ширина допоміжних проходів	$\geq 700$ мм
Відстань між двома столами, якщо між ними є стілець	$\geq 1300$ мм

У розглянутому приміщенні: відстань між двома столами становить 1500 мм, відстань між двома столами в ряді – 1500 мм, а між рядами – 4000 мм.

Отже, норми виконуються.

У технічних умовах роботи ЕОМ вказуються робочі діапазони параметрів мікроклімату:

- температура повітря 5 – 45 °С,
- відносна вологість повітря 40 – 90 %.

Однак, вимоги точного регулювання параметрів повітряного середовища приміщення значно звужують ці діапазони.

З метою забезпечення комфортних умов для персоналу, а також максимальної безвідмовності функціонування техніки, встановлюють вимоги до повітряного середовища приміщень. Так, у приміщенні повинні бути:

- температура повітря 18 – 22 °С,
- відносна вологість повітря 50 – 60 %,
- атмосферний тиск 1013 – 1013,5 гПа.

При зниженні тиску погіршується відвід тепла від елементів ЕОМ, знижуються ізоляційні властивості повітря.

Як було показано вище, показники об'єму і площі приміщення на одного працюючого відповідають нормативним значенням.

Роботи, що проводяться в приміщенні відносяться до легких фізичних робіт групи 1а, відповідно до ГОСТ 12.1.005-88, оскільки вони проходять сидячи і не вимагають фізичного навантаження, здійснюються при нормальних метеорологічних умовах і не викликають забруднення одягу і рук. Витрати енергії не перевищують 172 Дж/с (155 кКал/год).

Наведено норми температури, відносної вологості і швидкості руху повітря на робочих місцях відповідно до ГОСТ 12.1.005-88, що встановлює норми виробничого мікроклімату (таблиці 4.3 та 4.4). Дані приведені для приміщень з незначним надлишком явного тепла (до 20 кКал/год x м<sup>3</sup>) для виконання легких робіт.

Основними джерелами тепла у приміщенні є:

- сонячна радіація,
- система опалення,
- люди, що працюють у приміщенні,
- устаткування.

Таблиця 4.3 – Норми температури, відносної вологості і швидкості руху повітря на постійних робочих місцях

Період року	Норми	Температура повітря t, °С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодний	оптим.	22-24	40-60	менше, ніж 0,1
	доп.	21-25	менше 75	менше, ніж 0,1
Теплий	оптим.	23-25	40-60	0,1
	доп.	22-28	менше, ніж 55	0,1-0,2

Таблиця 4.4 – Відносна вологість повітря в теплий період року

Температура повітря t, °С	28	27	26	25	24	≤23
Відносна вологість, %	≥55	60	65	70	75	75

Приведено дані параметрів мікроклімату у приміщенні у грудні (таблиця 4.5).

Таблиця 4.5 – Результати виміру параметрів мікроклімату у приміщенні

Параметр	Значення
Температура повітря t, °C	18 – 22
Відносна вологість, %	59 – 65
Швидкість руху повітря, м/с	0,11

У розглянутому приміщенні значення параметрів мікроклімату відповідають нормативним (див. таблицю 4.5). Постійність цих параметрів підтримується загальною системою утеплення і кондиціонування повітря. При цьому використовується кондиціонер SAMSUNG AQT-24A5RE, а при необхідності здійснюється провітрювання приміщення. Наведено параметри кондиціонера SAMSUNG AQT-24A5RE (таблиця 4.6). Він забезпечує встановлені норми мікроклімату у приміщенні.

Таблиця 4.6 – Параметри кондиціонера SAMSUNG AQT-24A5RE

Параметр	Значення
Потужність охолодження	6.8 кВт
Продуктивність охолодження	24 000 БТЕ/год
Потужність обігріву	6,9 кВт
Продуктивність обігріву	24 000 БТЕ/год
Видалення вологи з повітря	3 л/год
Циркуляція повітря	14 м <sup>3</sup> /хв

Джерелами пилу у приміщенні є: книги, документація, роздруківки, а також одяг, взуття працівників і зовнішнє повітря.

Встановлений у приміщенні кондиціонер SAMSUNG AQT-24A5RE забезпечує встановлені норми чистоти поступаючого зі сторони приміщення

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

повітря, що надходить ззовні. У приміщенні періодично проводиться вологе прибирання. Зазначені умови забезпечують підтримку в нормі параметрів чистоти повітряного середовища.

У приміщенні використовується природне і штучне освітлення. Природне освітлення здійснюється з допомогою двох вікон загальною площею  $S=7 \text{ м}^2$ , що забезпечує коефіцієнт природної освітленості  $E=1,5\%$ . Це відповідає СНіП I-4-79.

Штучне освітлення у приміщенні здійснюється системою загального рівномірного освітлення, що реалізована на основі люмінесцентних ламп типу ЛДЦ-40-1, що мають наступні параметри:

- висока світловіддача;
- тривалий термін служби;
- мала яскравість освітлювальної поверхні;
- близькість спеціального складу до природного освітлення.

Робота за монітором ПЕОМ по розряду зорових робіт відноситься до III типу (роботи високої точності з розміром об'єкта 0,2-0,4 мм) [10]. При загальному освітленні, освітленість робочого місця повинна складати від 200 до 400 лк.

При штучному освітленні нормуються наступні параметри:

- $E$  (лк) – найменша припустима освітленість;
- $M$  – показник дискомфорту;
- $K_p$  (%) – коефіцієнт пульсації освітлення.

Номинальний світловий потік лампи білого свічення ЛДЦ-40-1:  $\Phi_{л}=3120 \text{ лм}$ .

У приміщенні застосовуються світильники, у яких встановлені дві лампи. Висота підвіски світильника визначається за формулою:

$$h = H - h_c - h_p - h_{п}, \quad (4.1)$$

де  $H$  – висота приміщення (м),

$h_c$  – висота світильника (м),

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		

$h_{\text{П}}$  – відстань від стелі до підвіски (м),

$h_{\text{Р}}$  – висота робочої поверхні (м).

Для розглянутого приміщення:

$H = 4$  м,

$h_{\text{С}} = 0,15$  м,

$h_{\text{П}} = 0$  м (підвісу немає),

$h_{\text{Р}} = 0,8$  м.

Звідси  $h = 4 - 0,15 - 0,8 = 3,05$ (м).

Світильники розташовані в 2 ряди. Висота підвіски світильників складає 3,05 м відносно підлоги, відстань між рядами 3 м, відстань від ряду до стіни 1,5 м. Приміщення має наступні габарити:

– довжина  $A = 8$  м,

– ширина  $B = 6$  м.

Визначимо освітленість у робочій точці. Для розрахунку загальної рівномірної освітленості при горизонтальній робочій поверхні використовуємо метод коефіцієнта використання світлового потоку.

Розрахункова формула для світлового потоку світильника має такий вигляд:

$$\Phi_{\text{л}} = \frac{E \cdot K_{\text{з}} \cdot S \cdot Z}{N \cdot n}, \quad (4.2)$$

де  $N$  – кількість світильників у приміщенні ( $N = 6 \cdot 2 = 12$ );

$n$  – коефіцієнт використання світлового потоку;

$\Phi_{\text{л}}$  – світловий потік ламп;

$K_{\text{з}}$  – коефіцієнт запасу ( $K_{\text{з}} = 1,5$ );

$Z$  – коефіцієнт нерівномірності;

$S$  – площа приміщення;

$E$  – освітленість, створювана усіма світильниками.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

Звідси одержуємо формулу для розрахунку освітленості на робочому місці:

$$E = \frac{\Phi_{л} \cdot N \cdot n}{K_{з} \cdot S \cdot Z} \quad (4.3)$$

Коефіцієнт використання світлового потоку залежить від:

- КПД кривої розподілу сили світла світильника;
- коефіцієнта відбивання стелі  $R_{п}$  і стін  $R_{с}$ ;
- висоти підвісу світильників  $h_{п}$ ;
- показника приміщення  $i$ :

$$i = \frac{A \cdot B}{h \cdot (A + B)} \quad (4.4)$$

Тобто  $i = (8 \cdot 6) / (3,05 \cdot (8 + 6)) = 1,12$ .

Стеля і стіни пофарбовані в білий колір.

Приймаємо  $R_{п} = 50\%$ ,  $R_{с} = 30\%$ .

Звідси  $n = 31\%$ ,

$$E = \frac{(3120 \cdot 2) \cdot 12 \cdot 0,31}{48 \cdot 1,1 \cdot 1,5} = 293 \text{ лк}.$$

Оскільки по розряду зорової роботи робота за дисплеєм ПЕОМ відноситься до III типу (високої точності, розмір об'єкта 0.2-14 мм), то при загальному висвітленні освітленість робочого місця повинна складати від 200 до 400 лк, рекомендована освітленість при роботі з дисплеєм ПЕОМ складає 200 лк, а при сполученні роботи з документами — 400 лк. Фактична освітленість на робочому місці складає 293 лк. Таким чином для роботи з дисплеєм цілком достатньо існуючих джерел світла, однак робота з документами повинна вестися при природному освітленні або за допомогою додаткових місцевих джерел освітлення.

У приміщенні основними джерелами шумів є: вентилятори системи охолодження ПЕОМ і кондиціонер. Згідно ГОСТ 12.1.003-83, нормованою

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						86
Змн.	Арк.	№ докум.	Підпис	Дата		

шумовою характеристикою робочих місць при постійному шумі являються рівні звукових тисків у децибелах в октавних смугах. Сукупність таких рівнів називається граничним спектром (ГС), номер якого дорівнює рівню звукового тиску в октавній смузі із середньо-геометричною частотою 1000 Гц. Приведено значення звукового тиску у приміщенні при роботі принтера (таблиця 4.7).

Таблиця 4.7 – Рівні звукового тиску в дБ на робочих місцях

	Середньо-геометричні смуги октавних частот							
	63	125	250	500	1000	2000	4000	8000
Норма	71	61	54	49	50	42	40	38
Лазерний принтер	56	56	58	61	57	60	60	60

Для розрахунку еквівалентного рівня шуму, створеного у приміщенні принтером і працюючими ПЕОМ, використаємо формулу:

$$L_{EKB} = \frac{1}{T} \cdot 10 \cdot \lg \left( \sum 10^{0,1 \cdot L_i} \right) \quad (4.5)$$

Враховуючи, що  $L_{\text{принтера}} = 60$  дБ і принтер працює в середньому 1 год в день, а  $L_{\text{ПЕОМ}} = 35$  дБ і ПЕОМ працює кожен робочий день, еквівалентний шум складає:

$$L_{EKB} = \frac{1}{8} \cdot 10 \cdot \lg \left( 10^{0,1 \cdot 60} + 10^{0,1 \cdot 35 \cdot 8} \right) = 35,78 \text{ дБ}. \quad (4.6)$$

По ГОСТ 12.1.003-83  $L_{\text{допустиме}} = 50$  дБА, відповідно, рівень шуму у приміщенні не перевищує норму.

Потенційну небезпеку для людини представляють електричні прилади і устаткування. Ураження людини електричним струмом може відбутися в результаті дотику до відкритих струмопровідних частин при ушкодженні

ізоляції мережевих шнурів, при пробої, при короткому замиканні або в результаті необережних дій самої людини.

Приміщення за ступенем небезпеки ураження електричним струмом відноситься до приміщень без підвищеної небезпеки. Споживачами електроенергії являються ПЕОМ, принтери і джерела освітлення.

Корпуси сучасних ПЕОМ виготовляються із пластмас (передня панель) і металу (верхня кришка і задня панель). При дотику до металевих частин корпусу ПЕОМ у випадку пробою на корпус людина може потрапити під небезпечну для життя напругу. Для запобігання цьому в конструкції ПЕОМ передбачене спеціальне електричне з'єднання з нульовим захисним провідником металевих частин корпусу, що можуть виявитися під напругою. Для цього в ПЕОМ застосовується спеціальна мережева вилка з трьома контактами (два контакти призначені для підключення живлення, а третій – для підключення до зануленого проводу).

Корпуси моніторів виготовляються з непровідних матеріалів, а живлення здійснюється спеціальним кабелем, що підключається до системного блоку ПЕОМ так, щоб виключити ураження людини електричним струмом.

Корпуси сучасних принтерів також виготовляються з пластмас, а конструкція кабелю живлення аналогічна кабелю ПЕОМ. Тому небезпека ураження струмом при дотику людини до корпусів принтера чи дисплея незначна.

Електропроводка у приміщенні розміщена в спеціальних захисних коробах.

У приміщенні застосовуються наступні засоби захисту:

- малі напруги;
- занулення неструмопровідних частин, що можуть виявитися під напругою;
- ізоляція струмопровідних частин;
- попереджувальні написи.



На випадок аварії передбачені запобіжники в приладах. До роботи в приміщення не допускаються особи, що не пройшли інструктаж по електробезпеці.

При вході у приміщення висить плакат з інструктажем по техніці безпеки при роботі та ремонті електроустаткування. Силовий щиток має такі запобіжні властивості, як система блокування відкривання кришки щитка та ізолюючий килимок.

При розробці документації до дипломного проекту використовувалась ПЕОМ типу AMD Athlon XP 1800+ (1533 MHz) і лазерний принтер HP LaserJet 1020.

Найбільш значимі фактори при роботі з ПЕОМ наступні:

- іонізоване випромінювання;
- низькочастотні електромагнітні випромінювання;
- зорова напруженість.

Шум у приміщенні, де виконуються роботи, потребуючі концентрації уваги, не повинні перевищувати 55 дБА за ГОСТ 12.1.003-83.

Джерелом шуму у приміщенні є принтер, рівні звуку якого по технічному паспорті не більш 63 дБА.

Джерелами рентгенівського і ультрафіолетового випромінювання у приміщенні є електронно-променеві трубки (екрани моніторів). Такі екрани генерують м'яке рентгенівське випромінювання з енергією фотонів від 1 КЕВ до 1 МЕВ і відносяться до категорії іонізованих випромінювань.

Захист від впливу цих випромінювань може бути досягнутий такими способами:

- шляхом віддалення на можливо максимальну відстань оператора від екрана (в основному 0,5...0,7 м);
- скорочення часу безперервної роботи (захист по часу);
- розміщенням оператора під деяким кутом до діагональної осі екрана.

Відповідно до ГОСТ 27016-86 для відеотерміналів на основі ЕПТ нормовані значення наступні:

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						89
Змн.	Арк.	№ докум.	Підпис	Дата		

– потужність дози рентгенівського випромінювання в точці простору на відстані 5 см від поверхні екрану монітора не повинна перевищувати 0.03 мкР/с при 41 годинному робочому тижні;

– щільність потоку ультрафіолетового випромінювання не повинна перевищувати 10 Вт/м<sup>2</sup>.

Приведено фактичні значення для моніторів, які використовуються (таблиця 4.8).

Таблиця 4.8 – Фактичні та нормативні значення характеристик дисплея

Найменування Параметрів	Нормовані значення	Фактичні значення
Розміри символів по висоті h, мм	$\geq 3$	4
Ширина лінії, мм	$\geq 0.4$	0.4
Яскравість зображення, лм	100	100
Потужність дози рентгенівського випромінювання на відстані 5 см,	$\leq 0.03$	0.01
Щільність потоку ультрафіолетового випромінювання, Вт/м <sup>2</sup>	$\leq 10$	8
Шум, дБА	$\leq 40$	10

Електромагнітні випромінювання низької частоти (від 12 до 150 Гц) роблять найбільш шкідливий вплив на організм людини. Тривалий вплив низькочастотних полів сприяє порушенню репродуктивної функції і виникненню раку.

Для зниження рівня перемінного електромагнітного поля в сучасних моніторах, що відповідають специфікаціям Low Radiation (LR), MPRII і TCQ92, застосовуються котушки компенсації, встановлені на електронно-променевої трубці (ЕПТ), а також спеціальні матеріали в її конструкції. Застосовувані при роботі у приміщенні монітори LG Studioworks 575C, 2008 року виготовлення, задовольняють встановлені норми.

В найбільшій мірі негативний вплив на зір при роботі з ПЕОМ зв'язаний з нерівномірно спроектованим освітленням, прямими і відбитими від екранів відблисками, несприятливим розподілом яскравості в полі зору, пульсацією екрана, неправильним розміщенням робочого місця відносно світлових променів.

Оптимальною для робочих приміщень, призначених для роботи з відеотерміналами, вважається освітленість 200 – 400 лк. Стрибок яскравості при зміні полів зору повинен бути мінімальним, тобто інтенсивність освітлення поверхні, де знаходяться рукописи і документи, не повинні перевищувати яскравості екрана дисплея. Співвідношення яскравості екрана і безпосередньо найближчого оточення не повинне перевищувати три до одного. Фактично дані вимоги на робочому місці виконуються згідно вимог ГОСТ 27016-86.

При тривалій роботі з друкувальним пристроєм вимоги по охороні праці в області тривалих шумових впливів на оператора виконуються і відповідають встановленим нормам.

#### 4.2 Пожежна безпека

Приміщення згідно ОНТП 24 відноситься до категорії В, класу П-Іа ПУЕ 76/87 по пожежній небезпеці. У приміщенні є займисті речовини:

- волокнисті (папір);
- тверді (дерево).

Пожежа у приміщенні представляє особливу небезпеку, оскільки пов'язана зі значними матеріальними втратами. Як відомо, пожежа може виникнути при взаємодії горючих речовин, окислювача і джерела запалювання. Горючими речовинами являються будівельні матеріали для акустичної обробки приміщення, перегородки, двері, підлога, папір для принтеру, корпуси ПЕОМ і принтерів, ізоляція кабелів. Особливістю сучасних ПЕОМ являється дуже висока щільність розміщення елементів електронних схем. При проходженні

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дата		

електричного струму по провідниках і деталях виділяється тепло, що в умовах їх високої щільності може привести до перегріву. Надійна робота окремих елементів і електричних схем в цілому забезпечується тільки у визначених інтервалах температури, вологості і при заданих електричних параметрах. Кабельні лінії зв'язку являються найбільш пожежонебезпечним місцем. Для зниження загоряння і здатності розповсюдження вогню кабелі покривають вогнетривким покриттям.

Для гасіння пожежі на початковій стадії її виникнення у приміщенні встановлені 3 вуглекислотних вогнегасники ОУ-2.

Для передбачення пожежі в приміщенні прийняті такі міри:

- передбачений вільний доступ до мережевих рубильників і вимикачів;
- на випадок короткого замикання передбачені запобіжники і автоматичне відключення мережі;
- в наявності є вогнегасники ОУ-2 для гасіння електрообладнання і ОХП-10 для гасіння об'єктів, що не знаходяться під напругою;
- вхідні двері приміщення відкриваються на зовні;
- ширина дверей не менше 0,8 м, а висота проходу більше 1 м;
- у приміщенні є план евакуації людей;
- у спільному коридорі, поруч з приміщенням, знаходиться пожежний кран;
- ширина загального коридору, ширина дверей, висота дверей відповідають нормативним значенням (таблиця 4.9).

Таблиця 4.9 – Характеристики евакуаційних виходів

	Нормативні значення, м	Існуючі значення, м
Ширина коридору	> 2,0	2,2
Ширина дверей	> 0,8	1,25
Висота дверей	> 2,0	2,14

## ВИСНОВКИ

При розробленні арифметичного пристрою RISC-процесора розглянуто загальні питання, пов'язані з призначенням, організацією та основними його характеристиками, а також приведено його класифікацію.

У роботі проведена декомпозиція складної задачі синтезу арифметичного пристрою RISC-процесора на окремі підзадачі; розроблена загальна структура. Детально пояснений принцип роботи в цілому.

Розроблено структурну схему арифметичного пристрою RISC-процесора та побудовано алгоритм його роботи. Розроблено функціональну схему та проведено функціональну симуляцію арифметичного пристрою RISC-процесора.

Проведено аналіз санітарно-гігієнічних умов праці та пожежної безпеки.

Результати роботи плануються до використання (Додаток Е).

					Ошибка! Источник ссылки не найден. ПЗ	Арк.
						93
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A High-Performance Computing Architecture. [Електронний ресурс] / Technical
2. Автоматизированное проектирование цифровых устройств / С. С. Бадудин, Ю. М. Барнаулов, В. А. Бердышев и др. – М.: Радио и связь, 1981. – 240 с.
3. Акаев А. А. Оптические методы обработки информации / А. А. Акаев, С. А. Майоров – М.: Высш. школа, 1988. – 237 с.
4. Анкудинов Г. И. Синтез структуры сложных объектов: логико-комбинаторный подход. / Г. И. Анкудинов – Л.: ЛГУ, 1986. – 260 с.
5. Балашов Е. Н. Эволюционный синтез систем / Е. Н. Балашов - М.: Радио и связь, 1985. – 328 с.
6. Бухштаб А. И. Электронные клавишные вычислительные машины / Бухштаб А. И., Каневский Е. А., Хохлов Л. М. – Л.: Энергия, 1974. – 160 с.
7. Гайченко В. А. Коваль Г. М. Основи безпеки життєдіяльності людини: Навч. посіб. — 2-ге вид. стереотип. — К.: МАУП 2004.
8. Глушков В. М. Автоматизация проектирования вычислительных машин / Глушков В.М., Капитонова Ю. В., Летичевский А. А. – Киев: Наукова думка, 1975. - 230 с.
9. Глушков В. М. и др. Вычислительные машины с развитыми системами интерпретации / Глушков В.М. – Киев: Наукова думка, 1970. – 260 с.
10. Желібо Є.П., Заверуха Н. М., Зацарний В.В., Безпека життєдіяльності/ за ред. Желібо Є.П. – К.: Каравела , 2003. – 328 с.
11. Закон України „ Про охорону праці” в редакції від 21 листопада 2002 р.
12. Интеллектуальные системы автоматизированного проектирования больших и сверхбольших интегральных схем: В. А. Мищенко, Л. М. Городецкий, Л. И. Гурский и др. - М.: Радио и связь, 1988. - 272 с.
13. Киносита К. Логическое проектирование СБИС / Киносита К., Асада К., Карацу О. – М.: Мир, 1988. - 309 с.
14. Методичні вказівки до написання розділу «Охорона праці» в дипломних проектах з освітньо-кваліфікаційного рівня «Спеціаліст» для спеціальності

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		94

- 7.091501 – Комп'ютерні системи та мережі / Укл. Г.В. Сапожник, Н.М. Васильків. – Тернопіль: ТАНГ, 2004. – 24 с.
- 15.Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Спеціаліст”. Спеціальність – “Комп'ютерні системи та мережі” /О.М. Березький, Н.М. Васильків, І.В. Васильцов, Р.Б. Трембач /Під ред. М.П. Карпінського.–Тернопіль: ТНЕУ, 2008. – 38с.
- 16.Мурога С. Системное проектирование сверхбольших интегральных схем: 2х кн. Кн. 1 / Мурога С. – М.: Мир, 1985. – 288 с.
- 17.Николаев В. И. Системотехника: методы и приложения / Николаев В. И., Брук В. М. – Л.: Машиностроение, 1985. – 199 с.
- 18.Озкарахан Э. Машины баз данных и управление базами данных / Озкарахан Э. – М.: Мир, 1989. - 696 с.
- 19.Основы теории вычислительных систем: Учеб. пособие / Под ред. С. А. Майорова. - М.: Высш. школа, 1978. - 408 с.
- 20.Проектирование цифровых вычислительных машин: Учеб. пособие для вузов / Под ред. С. А. Майорова. - М.: Высш. школа, 1972. - 344 с.
- 21.Савельев А. Я. Прикладная теория цифровых автоматов: Учеб. для вузов по спец.: ЭВМ. - М.: Высш. школа, 1987. - 272 с.
- 22.Самофалов К. Г. Цифровые ЭВМ. Теория и проектирование / Самофалов К. Г., Корнейчук В. И., Тарасенко В. П.- Киев: Вища школа, 1983. - 456 с.
- 23.СБИС для распознавания образов и обработки изображений: Под ред. К. Фу.- М.: Мир, 1988. - 248 с.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						95
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А. Арифметичний пристрій RISC-процесора. Схема структурна

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		96



Додаток Б. Арифметичний пристрій RISC-процесора. Схема функціональна

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		97

Додаток В. Арифметичний пристрій RISC-процесора. Схема електрична  
принципова

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						98
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Г  
Арифметичний пристрій RISC-процесора

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						99
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Д. Арифметичний пристрій RISC-процесора. Лістинг.

```
entity AU is port( CLK : in BIT;  
RST : in BIT;  
START : in BIT;  
RD: in BIT;  
WRD : in BIT;  
RET : in BIT;  
CALL: in BIT;  
DI : in BIT_VECTOR(15 downto 0);  
AB : in BIT_VECTOR(2 downto 0);  
AD : in BIT_VECTOR(2 downto 0);  
AQ : in BIT_VECTOR(2 downto 0);  
ARET : in BIT_VECTOR(12 downto 0);  
ACOP : in BIT_VECTOR(3 downto 0);  
RDY : out BIT;  
ARETO : out BIT_VECTOR(12 downto 0);  
DO : out BIT_VECTOR(15 downto 0);  
BO : out BIT_VECTOR(15 downto 0);  
CNZ: out BIT_VECTOR(2 downto 0));  
end AU;
```

Опис архітектури AU, в якому не приведені описи її компонентів, виглядає таким чином:

```
type STAT_AU is (free,mpu,mpyl);-- стани автомата  
signal st:STAT_AU;
```

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						100
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    signal      b,q,d,y,dp,aretc,aretc0:bit_vector(15      downto      0);      signal
c0,c15,csh,zlsm,wr,mult,outhl:bit;      signal      rdym,zmpy,nmpy:bit;      signal
cnzr,cnzo,cnzi:bit_vector(2 downto 0); begin
    U_FM: FM2 port map(CLK      -- блок регістрової пам'яті WR=>wr,
INCQ=>outhl, CALL=>CALL, AB=>AB,  AD=>AD,   AQ=>AQ
    ARETC=>aretc
    Q=>q,   B=>b,   D=>d, ARETCO=>aretc0);
    aretc<=cnzr&ARET; cnzo<=aretc0(15 downto 13); ARETO<=aretc0(12 downto 0);
    MUX_C:c0<='1' when ACOP(1 downto 0)="10"else --мультиплексор C0
    cnzi(2) when ACOP(1 downto 0)="01"else '0'; U_LSM:LSM port map(F=>ACOP(3
downto 2) -- LSM
    A=>d,    B=>b, C0=>c0,    Y =>y, C15=>c15, Z =>zlsm    ); U_MPU:MPU port
map(CLK,RST -- блок множення
    START=>mult,  OUTHL=>outhl
    DA=>d, DB=>b
    RDY=>rdym,   Z=>zmpy
    N=>nmpy, DP=>dp);
    MUX_CI:csh<=cnzi(2) when ACOP(1 downto 0)="01" else --зادвиг разр. cnzi(1)
when ACOP(1 downto 0)="11" else '0';
    ----- мультиплексор результату-----
    MUX_Q:q<=dp when st/=free else --результат множення
    csh&d(15 downto 1) when ACOP="1000" -- зрушення управо or ACOP="1001"
or ACOP="1011" else
    DI when WRD='1' else --вхідне дане
    d when RD='1' else --дане з FM за адресою Ad
    y; --результат LSM
    SR: process(CLK,RST) --регістр стану з мультиплексором
begin

```

										Арк.
										101
Змн.	Арк.	№ докум.	Підпис	Дата	Ошибка! Источник ссылки не найден. ПЗ					

```

if RST='1' then
cnzi<="000"; elsif CLK='1' and CLK'event then if RET='1' then
cnzi<=cnzo; elsif st=mpyl then
cnzi<='0'&nmpy&zmpy; elsif mult='0' then
cnzi<=c15&y(15)&zlsm;
103
end if; end if; end process;
mult<='1' when ACOP="1100" else '0';--дешифрація множення FSM_AU:
process(CLK,RST) --автомат управління
begin
if RST='1' then
st<=free; -- регістр стану автомата
elsif CLK='1' and CLK'event then
case st is
when free => if START='1'and mult='1'then --вільний st<=mpy; end if;
when mpy=> if rdym='1' then -- йде множення st<=mpyl ; end if;
when mpyl=> st<=free; --кінець множення end case; end if; end process;
--функції виходів автомата outhl<='1' when st=mpyl else '0';
wr<='1' when WRD='1' or st=mpyl or (st=mpy and rdym='1')
or (START='1' and mult='0') else '0'; RDY<='1' when st=mpyl or (WRD='0' and
st/=mpy and mult='0') else'0'; DO<=q; --вихідне дані BO<=B;
CNZ<=cnzi; --вихід регістра стану
end BEH;

```

#### 4.7.4 Випробувальний стенд для AU

AU є складним блоком з внутрішньою пам'яттю. Для його перевірки пропонується випробувальний стенд на основі простого пристрою мікропрограмного управління. Його опис архітектури (окрім опису AU як компоненту і сигналів -входів-виходів ALU) представлений нижчим.

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						102
Змн.	Арк.	№ докум.	Підпис	Дата		

```

type MICROINST is record -- формат мікрокоманды/
  ACOP:bit_vector(3 downto 0);      -- код операції AU AQ,AD,AB:bit_vector(2
downto 0); - -адреса FM DI:BIT_VECTOR(15 downto 0);      -- вхідне дані
  START,WRD,RD:bit; -- биты/ управління
end record;

constant n: positive:=6; --число мікрокоманд
type MICROPROGR is array(0 to n-1) of MICROINST;
constant mp:MICROPROGR:= ( -- ПЗП тестуючої мікропрограмы/
  ("0000","001","000","000",X"4000",'0','1','0'), --L 1,#4000hi
  ("0000","010","000","000",X"cfff", '0','1','0'), --L 2, #-3001hi
  ("0010","011","001","010",X"0000",'1','0','0'), --ADDINC 3,1,2
  ("0101","011","001","010",X"0000",'1','0','0'),      --SUBC      312
  ("1100","100","011","010",X"0000",'1','0','0') --MUL 4,3,2
  104
  104
  ("0000","000","100","000",X"0000",'0','','0','1')); -S 4 signal maddr:natural; begin
  CLK<=not CLK after 5 ns; -- генератор синхросигналу RST<=T,'0 after 25 ns; --
генератор сигналу скидання CTM:process(CLK,RST) begin -- лічильник мікрокоманд
  if RST=T then
    maddr<=0; elsif CLK=T and CLK'event then
      if (RDY=T and START=T) or WRD=T or RD='1 then
        maddr<=(maddr+1) mod n; -- +1 до лічильника end if; end if; end process;
      ROM_MP:(ACOP,AQ,AD,AB,DI,START,WRD,RD)<=mp(maddr); UUT : AU
port map (CLK,RST,      -тестируемое Аї
  START => START
  RD => RD, WRD => WRD
  RET => RET,      CALL => CALL
  DI => DI

```

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		103

AB => AB, AD => AD, AQ => AQ

ARET => ARET, ACOP => ACOP

RDY => RDY, ARETO => ARETO

DO => DO, BO=>BO, CNZ => CNZ); end TB\_ARCHITECTURE;

					<b>Ошибка! Источник ссылки не найден. ПЗ</b>	Арк.
						104
Змн.	Арк.	№ докум.	Підпис	Дата		



MorozovaKSMc-51