

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії

ЛЕСИК Григорій Васильович

**Багатоканальні засоби обміну з буферизацією  
даних для систем управління з промисловою  
мережею / Multichannel Means of Exchange of  
Buffering Data for Control Systems with Industrial  
Network**

спеціальність: 123 - Комп'ютерна інженерія  
магістерська програма - Комп'ютерна інженерія

Магістерська робота

Виконав студент групи КІм-21  
Г. В. Лесик

Науковий керівник:  
д.т.н., професор І. Г. Цмоць

Магістерську роботу допущено  
до захисту:

"31" 01 2018 р.

Завідувач кафедри

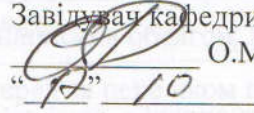
О. М. Березький

ТЕРНОПІЛЬ - 2018

Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії  
Освітній ступінь «магістр»  
спеціальність: 123 - Комп'ютерна інженерія  
магістерська програма - Комп'ютерна інженерія

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

  
О.М. Березький  
"12" 10 2016 р.

### **ЗАВДАННЯ НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**



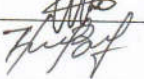

Лесика Григорія Васильовича

(прізвище, ім'я, по батькові)

1. Тема магістерської роботи «Багатоканальні засоби обміну з буферизацією даних для систем управління з промисловою мережею / «Multichannel data exchange and buffering data for industrial network management systems»  
керівник роботи д.т.н., професор І.Г. Цмоць  
затверджені наказом по університету від 11 жовтня 2016 р. № 669.
2. Строк подання студентом роботи «15» січня 2018 року
3. Вихідні дані до магістерської роботи  
Об'єкт дослідження – багатоканальні засоби обміну з буферизацією даних для систем управління з промисловою мережею.  
Предмет дослідження – пристрої обміну з буферизацією даних.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
  - Проаналізувати засоби обміну для штучних нейронних мереж .
  - Засоби реалізації пристроїв обміну .
  - Розробка пристрою обміну на базі багатопортової пам'яті для нейромереж.
  - Розробка структури пристрою обміну на базі багатопортової пам'яті для нейромереж .
  - Розробка алгоритму роботи пристрою обміну на базі багатопортової пам'яті для нейромереж.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
  - Структурна схема пристрою .
  - Блок-схема контролера обміну.
  - Функціональна схема контролера обміну.

– 7. Консультанти по роботі.

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Антиплагіат	Мельник Г.М., доцент		
Нормо-контроль	Гураль І. В., викладач		

7. Дата видачі завдання «13» жовтня 2016 р.

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Приміт
1	Огляд літературних джерел	3.11.2016 – 1.01.2017	Виконано
2	Проектування засобів обміну з буферизацією даних для систем управління з промисловою мережею	2.01.2017 – 31.05.2017	Виконано
3	Розробка концептуальної моделі багатопортової пам'яті	1.06.2017 – 25.01.2018	Виконано
4	Нормоконтроль, попередній захист	16.01.2018 – 2.02.2018	Виконано
5	Захист	4.02.2018	В

Студент

Лесик Г. В.

Керівник магістерської роботи

д.т.н., професор, І. Г. Цмоць

## ЗМІСТ

Вступ.....	9
1. Огляд літературних джерел.....	12
1.1 Характеристика багатопортової пам'яті.....	12
1.2. Огляд пам'яті для комп'ютерних систем реального часу.....	15
1.3 Огляд основних компонентів паралельної пам'яті.....	18
1.4. Огляд базових структур паралельної пам'яті.....	21
1.5 Огляд елементної бази для синтезу запам'ятовуючих середовищ.....	26
1.7 Огляд структур запам'ятовуючих середовищ.....	33
2. Проектування засобів обміну з буферизацією даних для систем управління з промисловою мережею.....	37
2.1. Розробка дерева цілей.....	37
2.2. Аналіз способу доступу до пам'яті.....	39
3.3. Аналіз способу побудови запам'ятовуючого середовища.....	43
2.4. Аналіз способу звертання до запам'ятовуючого середовища.....	47
2.5. Аналіз комутуючої мережі.....	52
2.6. Аналіз методів управління доступом до паралельної пам'яті.....	56
3. Розробка концептуальної моделі багатопортової пам'яті.....	69
3.2. Розробка структури багатопортової пам'яті.....	74
3.3. Розробка компонентів багатопортової пам'яті.....	75
3.4. Синтез багатопортової пам'яті для комп'ютерних систем реального часу. .....	80
3.6. Моделювання компонентів БПП.....	84
3.7. Розробка програми прикладу використання БПП в паралельно- потоківих системах.....	89
Висновки.....	95
Список використаних джерел.....	96



## ВСТУП

**Актуальність теми.** При розв'язанні значної частини задач в комп'ютерних системах, від обчислювальних засобів вимагається обробка в реальному масштабі часу потоків даних великої інтенсивності. При цьому можуть виникати конфлікти, коли декілька пристроїв будуть намагатись отримати доступ до пам'яті за один командний цикл. Основним шляхом забезпечення даних вимог є використання багатопортової пам'яті (БПП), що забезпечує вирішення всіх конфліктів, що можуть виникнути при взаємодії комп'ютерних систем. Створення пристрою обміну для комп'ютерних систем на базі БПП забезпечує паралельний доступ до множини даних і підтримку швидкісного обміну між елементами системи. БПП зводить до мінімуму проблеми, пов'язані з обміном даних, синхронізацією роботи компонентів та системи загалом. Організація обміну на основі БПП використовує принцип часового розподілу ресурсів пам'яті між пристроями, що під'єднуються до неї. Безконфліктний паралелізм обміну між пристроями висуває свої вимоги до пристроїв обміну, які насамперед пов'язані з необхідністю забезпечення швидкісного паралельного доступу до множини даних і високої ефективності використання обладнання. Зокрема необхідно забезпечити узгодження інтенсивності надходження даних з інтенсивністю доступу до пристрою обміну, можливість зміни ширини та часу доступу до пристрою обміну, роботу з різними за швидкодією зовнішніми пристроями і буферною пам'яттю, паралельне ввімкнення необхідної кількості зовнішніх пристроїв та виділення кожному зовнішньому пристрою періодичних фіксованих часових інтервалів доступу до пам'яті.

Особливістю БП є незалежні множини шин адреси та даних, що дозволяє за один цикл виконання команди збільшити кількість звертань до пам'яті. БП підтримує одночасний доступ до неї з усіх портів, кількість яких відповідає кількості шин адреси і даних. Така пам'ять забезпечує паралельний доступ до множини даних і вирішення усіх конфліктів, що виникають при такому доступі.

Реалізація такої пам'яті вимагає великих апаратних витрат, які насамперед залежать від паралельності доступу, а також від ємності пам'яті.

Тому актуальною задачею є розробка багатопортової пам'яті для комп'ютерних систем реального часу.

**Мета і задачі дослідження.** Метою роботи є вибір основних принципів побудови, що дозволять зменшити апаратні витрати, розробка структури та компонентів багатопортової пам'яті, розробка алгоритмів роботи і реалізація прикладу використання багато портової пам'яті реального часу.

Для досягнення поставленої мети в роботі розв'язуються такі задачі:

- огляд пам'яті для комп'ютерних систем реального часу, основних компонентів та базових структур паралельної пам'яті та огляд елементної бази для синтезу запам'ятовуючих середовищ.

- розробка дерева цілей та проведення системного аналізу;
- розробка концептуальної моделі багато портової пам'яті;
- формування вимог і вибір принципів побудови багатопортової пам'яті;
- розробка структури багато портової пам'яті;
- розробка компонентів багатопортової пам'яті: пристрою керування, контролера та генератора адрес багатопортової пам'яті;
- синтез багатопортової пам'яті для комп'ютерних систем реального часу.
- моделювання компонентів багато портової пам'яті на VHDL
- розробка програми прикладу використання багатопортової пам'яті в паралельно-потоківих системах.

**Наукова новизна отриманих результатів.** На основі виконаних досліджень розроблена структура та компоненти багатопортової пам'яті, здійснено моделювання компонентів багато портової пам'яті на VHDL та програма прикладу використання багатопортової пам'яті в паралельно-потоківих системах.

При цьому отримані такі нові результати:

- вибрано принципи побудови багато портової пам'яті, що дозволяють забезпечити вимоги до пам'яті, покращити її характеристики та зменшити обсяг використовуваної апаратури;

- розроблено структуру багато портової пам'яті на базі НВІС пам'яті з великим об'ємом та малим часом циклу запису і читання ;

### **Практичне значення отриманих результатів.**

- використання прийнятих принципів побудови багато портової пам'яті дозволяє забезпечити вимоги до пам'яті при розробці комп'ютерних систем реального часу, покращити характеристики пам'яті та зменшити обсяг використовуваної апаратури;

- при використанні розробленої структури контролера БПП забезпечується налаштування за допомогою зовнішніх тактових імпульсів на роботу з різними за швидкістю ОЗП і зовнішніми пристроями, програмне налаштування на генерацію необхідної послідовності адрес, мінімальний час вибірки даних з пам'яті, нарощування кількості пристроїв, що підключаються до ОЗП ;

- використання результатів моделювання компонентів багато портової пам'яті на VHDL дозволяє дослідити значення сигналів в часі;

- програма прикладу використання багатопортової пам'яті в паралельно-потоківих системах змоделювати роботу паралельно-потоківих системи і візуально побачити яке значення міститься в кожному такті на кожній сходинці системи, яка представлена багато портовою пам'яттю.



## 1. ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ

### 1.1 Характеристика багатопортової пам'яті.

У роботі для розв'язання задач комп'ютерних систем реального часу необхідно розробити багато портову пам'ять (БПП). Однією із головних переваг БПП є її гнучкість та простота збільшення або зменшення кількості пристроїв (портів), що підключаються до неї.

Основним параметром БПП є період звертання ( $T_z$ ) до неї пристроїв, який залежить від кількості пристроїв і від часу  $t_{\text{ц}}$  циклу читання (запису) в ОЗП. Даний час для пристроїв з однаковою швидкістю виконання операцій вводу-виводу визначається виразом  $T_z \geq t_{\text{ц}} \times n$ . Коли до БПП підключаються пристрої з різною швидкістю, то період звертання для  $L$ -швидкодійних пристроїв будуть  $T_{z_{\text{ш}}} \geq (L+1) \times t_{\text{ц}}$ , а для повільних -  $T_{z_{\text{п}}} \geq T_{\text{ш}} \times (n-L)$  [7].

Також при розробці БПП необхідно враховувати: витрати обладнання на реалізацію, кількість пристроїв між якими здійснюється обмін, швидкодію БПП, розрядність даних та частоту надходження даних, а також технічні та економічні вимоги і обмеження.

Одним з найважливіших показників БПП є швидкодія. Для кількісної оцінки швидкодії зазвичай використовують три параметри:

- час доступу, що відповідає інтервалу часу від моменту надходження адреси до моменту, коли дані заносяться в пам'ять або стають доступними;
- період звертання до пам'яті, що позначає мінімальний час між двома послідовними звертаннями до пам'яті. Період звертання включає в себе час доступу плюс деякий додатковий час. Додатковий час може вимагатися для загасання сигналів на лініях;
- швидкодія передачі - це швидкість, з якою дані можуть передаватися в пам'ять або з пам'яті.

Вибір структури БПП здійснюється за інтегральним параметром ефективності використання обладнання, що пов'язує витрати обладнання, ємність та інтенсивність доступу:

$$E = \frac{m_{БП} n_{БПi} Q}{W_{БП} t_{цБП}} \quad (1.1)$$

де  $m_{БП}$  - ширина доступу до БП,  $n_{БПi}$  - розрядність і-го каналу надходження даних,  $Q$  - ємність,  $W_{БП}$  - витрати обладнання на реалізацію БПП,  $t_{цБП}$  - час доступу до БПП.

Для ефективної реалізації БПП необхідно узгодити інтенсивність надходження даних з інтенсивністю доступу до БПП.

Інтенсивність надходження даних  $P_d$  має бути менша або дорівнювати інтенсивності доступу до БПП  $P_{БП}$ .

$$P_d \leq P_{БП} \quad (1.2)$$

Інтенсивність надходження даних залежить від частоти надходження даних, кількості та розрядності каналів надходження даних:

$$P_d = F_d \cdot m \cdot n_k \quad (1.3)$$

де  $F_d$  - частота надходження даних.;  $m$  - кількість каналів надходження даних;  $n_k$  - розрядність каналів надходження даних.

Інтенсивність доступу до БПП залежить від розрядності, частоти надходження даних в кожному каналі та від кількості каналів доступу до БПП.

$$P_{БП} = \sum_{i=1}^m n_{БПi} \cdot F_{БПi} \quad (1.4)$$

де  $n_{БПi}$  - розрядність і-го каналу надходження даних;  $F_{БПi}$  - частота надходження даних в і-му каналі,  $m$  - кількість каналів доступу до БПП.

При проектуванні пам'яті сучасних паралельних КС необхідно усунути вказані недоліки та забезпечити відповідність пам'яті наступним **вимогам**:

- мати організацію, що орієнтована на роботу з векторами і матрицями даних;
- забезпечити одночасне введення вхідного масиву даних і виведення раніше прийнятого масиву, дані в якому розміщені у відповідності до

алгоритму розв'язання задачі;

- забезпечувати введення і виведення даних як з однаковою, так і з різною тактовою частотою;
- виконувати функції переставляння та затримки даних на необхідне число тактів;
- забезпечувати багатоканальне введення та виведення даних, причому число входів і виходів може бути різним;
- бути адаптованою до структури даних і специфіки задач, які розв'язуються.

## 1.2. Огляд пам'яті для комп'ютерних систем реального часу

В структурах КС присутні три основні компоненти: пам'ять, операційні пристрої (процесори) і магістралі. Аналіз зв'язків між пам'яттю і операційними пристроями дозволив виділити три базові структури пам'яті, які використовуються в КС :

- пам'ять з послідовним доступом до елементів даних, в якій реалізується паралельна вибірка  $n$  розрядів даних (рисунок 1.1а);
- пам'ять з паралельним доступом до множини елементів даних, в якій паралелізм елементів даних реалізується на основі використання множини незалежних блоків пам'яті першого типу (рисунок 1.1б);
- пам'ять з паралельним доступом до множини елементів даних, що забезпечує паралельну адресацію і вибірку даних із загального поля пам'яті (рис.2.1в).

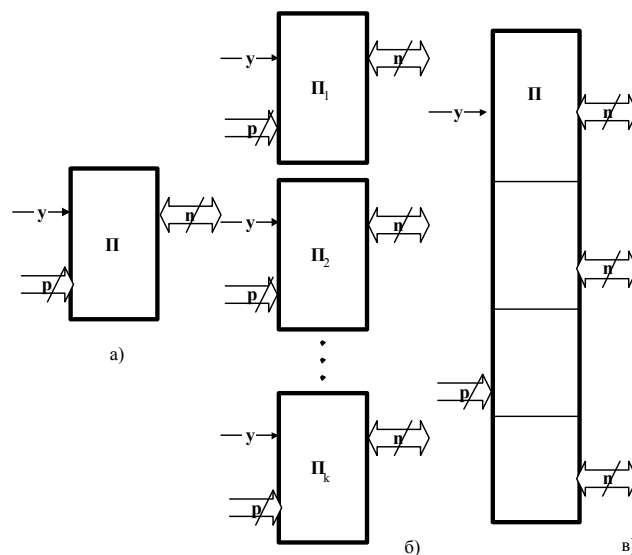


Рисунок 1.1. Базові структури пам'яті КС:

а) з послідовним доступом;

б) з паралельним доступом до множини незалежних блоків;

в) з паралельним доступом до загального поля пам'яті

На рисунку 1.1 -  $y$ ,  $n$  і  $p$  є відповідно шини управління, даних і адреси. В основу організації першої базової структури (рисунок 1.1а) пам'яті покладені

принципи одновимірності пам'яті та лінійності її адресного простору. Для даної пам'яті мінімальним неподільним елементом даних, що адресуються, є  $n$ -розрядне слово[1].

Друга базова структура пам'яті (рисунок 1.1б) реалізується на  $K$  блоках пам'яті першого типу, кожен з яких має власне запам'ятовуюче середовище з незалежною адресацією. Така структура пам'яті дозволяє реалізувати одну із наступних часових стратегій звертання до блоків:

- послідовну;
- з перекриттям у часі;
- паралельну.

Послідовна стратегія доступу до пам'яті передбачає повне розмежування процедури звертань до окремих блоків пам'яті в часовому просторі.

Доступ до блоків пам'яті з перекриттям у часі характеризується частковим перекриттям в часі процедури звертань до блоків пам'яті. При читанні з пам'яті час, необхідний для отримання даних з комірки, можна розбити на два інтервали: перший - безпосередньо доступ до ядра пам'яті та до даних в комірці, другий - передача даних на вихід. Такій доступ здійснюється в два етапи - поки дані з попереднього циклу читання одного блоку передаються на зовнішню шину даних, відбувається запит на читання з іншого блоку пам'яті. Таким чином, два цикли читання перекриваються в часі.

При паралельному методі доступу до окремих блоків пам'яті необхідно забезпечити одночасне встановлення фізичного зв'язку з множиною зовнішніх абонентів і організувати загальне поле пам'яті для даних абонентів.

Особливістю третьої базової структури пам'яті (рис.2.1в) є загальне поле пам'яті, що забезпечує паралельний доступ до множини елементів даних. Всередині такої пам'яті вирішуються всі конфлікти, що пов'язані з встановленням фізичного зв'язку з абонентом і необхідністю доступу до даних, що довільно адресуються. В даній структурі всі питання, що пов'язані з

паралелізмом доступу, вирішуються в самій пам'яті. Пам'ять, що відповідає даним вимогам, є паралельною.

Один з напрямків підвищення швидкодії КС є широке використання паралельної пам'яті. Але на даний момент створення ієрархічної пам'яті, у якій принципи паралелізму реалізовані на всіх рівнях, є недоцільне, що пояснюється кількома причинами. Перша - висока вартість і великі апаратні затрати. Друга причина - відсутність зовнішньої пам'яті паралельного типу. Крім цього, велику роль відіграє і те, що вартість паралельної пам'яті в більшій мірі залежить від паралельності доступу, ніж від ємності.

Раціональним на теперішній час бачиться організація пам'яті систем цифрової обробки сигналів ієрархічною багаторівневою на базі пристроїв з паралельним і послідовним доступом (рисунок 1.2), причому оперативну пам'ять, кеш і регістрову пам'ять доцільно реалізовувати у вигляді паралельної пам'яті [13].

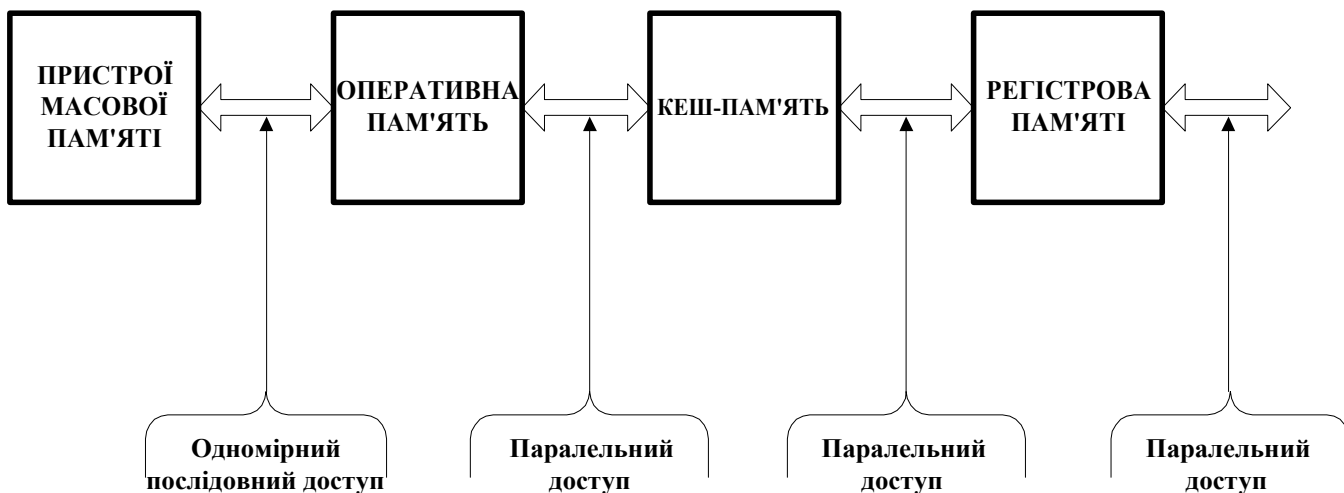


Рисунок 1.2. Ієрархічна багаторівнева структура пам'яті КС

### 1.3 Огляд основних компонентів паралельної пам'яті

Класифікувати паралельну пам'ять можна провести за способом розміщення і пошуком інформації (адресні, безадресні), за способом побудови запам'ятовуючого середовища (на основі множини модулів пам'яті, на основі одномірної спільної пам'яті), за способом звертань до запам'ятовуючого середовища (з послідовно-паралельним перетворенням, з паралельно-послідовним перетворенням).

За структурною організацією паралельна пам'ять дуже подібна до звичайної класичної послідовної пам'яті. Основними компонентами паралельної пам'яті є [2] :

- запам'ятовуюче середовище;
- адресний блок;
- комутуюча мережа;
- блок управління.

Запам'ятовуюче середовище є основою паралельної пам'яті , воно в значній мірі визначає основні параметри пам'яті: швидкодію, ємність, ширину і варіанти паралельного доступу. Запам'ятовуюче середовище паралельної пам'яті може бути двох видів:

- одновимірне, на базі спільної пам'яті;
- багатовимірне, на основі множини модулів пам'яті.

Для роботи з багатовимірними даними необхідно забезпечити формування багатовимірного запам'ятовуючого середовища, в якому будь-який елемент даних  $E$  задається своїми просторовими координатами (точкою в просторі)  $P_1, P_2, \dots, P_k$ , де  $k$  – розмірність простору. З множини елементів даних в декартовій системі координат утворюються точкові системи (решітки), які є моделлю запам'ятовуючого середовища. На рисунку 1.3 показано графічне зображення одно-, дво- та тривимірної точкової системи, що є моделями відповідно одно-, дво- та тривимірного запам'ятовуючого середовища.

При реалізації запам'ятовуючого середовища на базі множини модулів пам'яті загальна ємність паралельної пам'яті утворюється як сума ємностей пам'ятей всіх модулів. В такому запам'ятовуючому середовищі організація модулів пам'яті повинна забезпечувати можливість формування необхідної віртуальної конфігурації загального поля пам'яті. При цьому ключовою проблемою є вибір правил розміщення елементів даних в модулях пам'яті, які мають забезпечувати безконфліктність паралельної вибірки даних. Проблема безконфліктності пов'язана з необхідністю одночасної вибірки всіх елементів даних, що складають паралельну структуру.

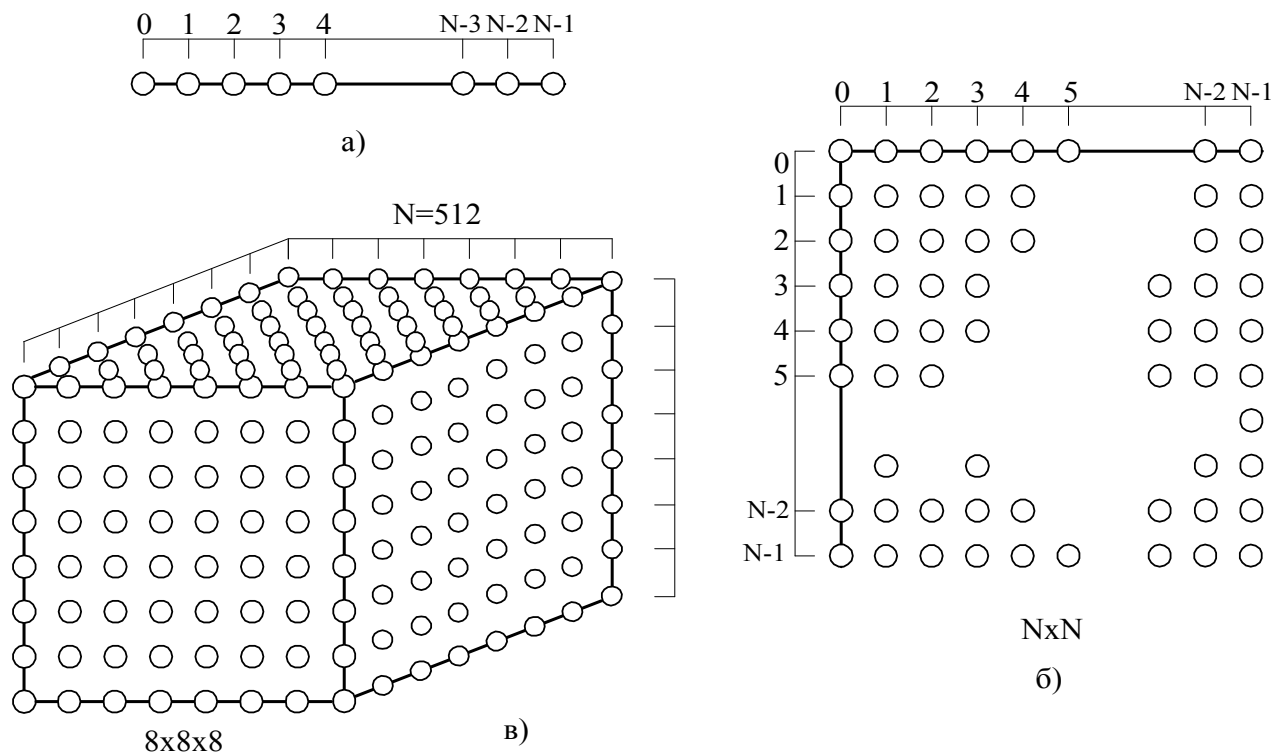


Рисунок 1.3. Моделі запам'ятовуючого середовища: а) – одновимірного; б) – двовимірного; в) – тривимірного

Використання одновимірної спільної пам'яті як запам'ятовуючого середовища вирішує всі проблеми, що пов'язані з безконфліктністю одночасної паралельної вибірки елементів даних. Як спільну пам'ять можна



використовувати звичайну класичну пам'ять, або розробляти спеціалізовану пам'ять[5].

Адресний блок реалізує функції формування і генерації необхідної послідовності адреси для запам'ятовуючого середовища. В паралельній пам'яті на базі множини модулів пам'яті, вхідна адреса перетворюється в множину адрес на входах модулів. Ці адреси, як правило, різні для різних модулів і обчислюється адресним блоком паралельної пам'яті. Крім того, такий блок здійснює обчислення, що пов'язані з структурними характеристиками паралельної пам'яті (розмірністю, числом модулів, набором геометричних форматів, організацією запам'ятовуючого середовища). В паралельній пам'яті на базі одномірної спільної пам'яті формування адреси здійснюється дещо простіше.

Комутуюча мережа паралельної пам'яті забезпечує необхідне з'єднання між пам'яттю та процесорами. Конкретна комутація при звертанні до паралельної пам'яті визначається алгоритмом розв'язання задачі. Комутуюча мережа є універсальною і забезпечує всі можливі переключення шин даних. Виконання таких переключень вимагає досить складного управління, в деяких випадках застосовується розділення мережі на блоки комутації запису і читання. Вибір конкретного варіанту реалізації комутуючої мережі в першу чергу залежать від конкретних алгоритмів упорядкування даних.

Блок управління формує сигнали керування для запам'ятовуючого середовища, здійснює налаштування генераторів адреси на формування необхідної послідовності адрес та комутуючої мережі на необхідні з'єднання між запам'ятовуючим середовищем і пристроями, які звертаються до пам'яті [10]. Крім того блок управління забезпечує вирішення конфліктів, що виникають при одночасному звертанні двох або більше пристроїв до одного модуля пам'яті (багатовимірне запам'ятовуюче середовище) або до однієї комірки (одновимірне запам'ятовуюче середовище). Блок управління може бути реалізований програмним, мікропрограмним або апаратним шляхом. На практиці в більшості випадків для реалізації функцій керування

використовуються комбіновані підходи з перевагою одного з перерахованих засобів. Перевалювання того чи іншого засобу визначається вимогами, які ставляться до блоку управління і до паралельної пам'яті за швидкодією та апаратними затратами.

#### 1.4. Огляд базових структур паралельної пам'яті

Структура паралельної пам'яті (ПП) може бути подана наступним чином:

$$S_{ПП} = \langle l, r, a, m \rangle,$$

де  $l$  – сукупність модулів пам'яті;  $r$  – спосіб комутації модуль пам'яті - зовнішній пристрій;  $a$  – спосіб адресації пам'яті ;  $m$  – ширина доступу (кількість входів-виходів) до пам'яті.

Спосіб комутації  $r$  модуль пам'яті - зовнішній пристрій та маршрутизацію даних забезпечує комутуюча мережа [15]. Характеристики комутуючої мережі в значній мірі впливають на характеристики ПП. У більшості випадків для з'єднань між модулями пам'яті та зовнішніми пристроями використовується наступні комутаційні мережі: з часовим розподілом ресурсів; просторовим розділенням; матричні; багатоярусні. Одними із основних параметрів таких мереж є пропускна здатність і можливості з'єднання. Комутаційні мережі можуть забезпечувати довільні варіанти комутації входів з виходами або обмежуються кількома варіантами..

За способом адресації комірок а ПП можна розділити на адресну, асоціативну та безадресну. В адресній ПП звертання до комірки пам'яті здійснюється у відповідності з адресою  $[A_i] \rightarrow D_i$ , де  $A_i$  – адреса комірки,  $D_i$  – вміст комірки пам'яті за адресою  $A_i$ . Вибірка даних з асоціативної ПП здійснюється за певними ознаками, а з безадресної - шляхом звертання до попередньо визначеної комірки пам'яті з наступним переміщенням даних.

В залежності від алгоритмів розв'язання задач та вимог застосування можна синтезувати багато структур паралельної пам'яті (ПП), які відрізняються як способами побудови запам'ятовуючого середовища, комутуючої мережі, розміщенням і пошуком інформації, так і технічними параметрами. Задача опису та розробки різноманіття можливих структур ПП є складною, доцільною є розробка і дослідження базових структур паралельної пам'яті, на основі яких можна було б синтезувати пам'ять для конкретних застосувань.

Аналіз структур паралельної пам'яті [9] дозволив виділити три базові структури паралельної пам'яті:

- з часовим розподілом ресурсів пам'яті;
- на основі множини модулів пам'яті;
- на основі багатовходових комірок пам'яті.

Перша базова структура ПП з часовим розподілом ресурсів пам'яті наведена на рисунку 1.5, де БлА- блок адреси; ПК – пристрій керування. Основним елементом такої ПП є модуль пам'яті, який є звичайною класичною пам'яттю. При такій організації ПП зв'язок між зовнішніми пристроями та модулем пам'яті здійснюється через комутатор мережі. В більшості випадків для доступу до магістралі модуля пам'яті використовується алгоритм фіксованих часових інтервалів [8]. Цей алгоритм надає кожному пристрою циклічно однаковий часовий інтервал  $t_{ц}$  незалежно від того чи пристрою необхідний доступ до пам'яті чи ні. Мінімальний час доступу до магістралі модуля пам'яті  $t_{ц}$  визначається часом циклу читання (запису) з пам'яті. Реалізувати такий доступ до шини пам'яті дозволяє ПК, який синхронізує доступ до магістралі модуля пам'яті асинхронно працюючих зовнішніх пристроїв. В такій структурі використовується паралельно-послідовне перетворення, яке називається масштабуванням паралелізму доступу до спільного ресурсу – пам'яті. Таке масштабування широко використовується для погодження ширини доступу  $m$  з швидкістю обміну  $V$ . Максимальна швидкість  $V$ , з якою зовнішні пристрої можуть обмінюватись з пам'яттю дорівнює:

$$V = \frac{1}{m \times t_{\text{ц}}}$$

Однією з основних переваг такої структури є гнучкість і простота збільшення або зменшення кількості пристроїв, що підключаються до магістралі модуля пам'яті. Апаратні витрати на реалізацію ПП з часовим розподілом ресурсів становлять:

$$W_1 = W_{\text{модП}} + mW_{\text{БлА}} + W_{\text{ПК}} + W_{\text{КМ1}},$$

де  $W_{\text{модП}}$ ,  $mW_{\text{БлА}}$ ,  $W_{\text{ПК}}$ ,  $W_{\text{КМ1}}$  - апаратні витрати відповідно на реалізацію модуля пам'яті, блока адреси, пристрою керування та комутуючої мережі з часовим розподілом ресурсів.

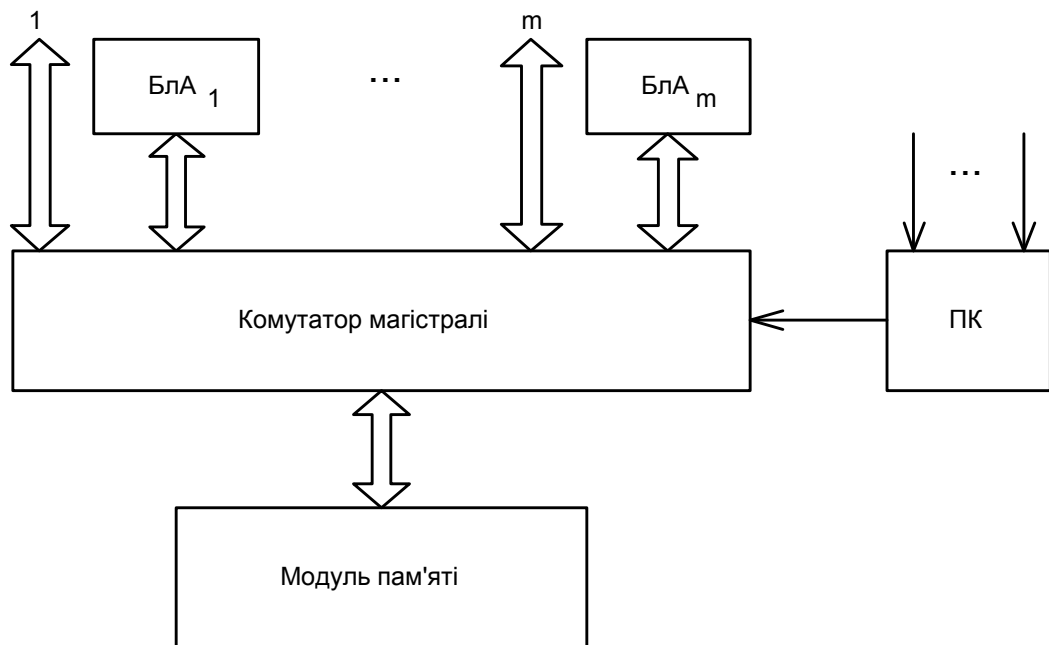


Рисунок 1.4. Базова структура паралельної пам'яті з часовим розподілом ресурсів пам'яті

Друга базова структура паралельної пам'яті на основі множини модулів пам'яті наведена на рисунку 1.5, де БУ - блок управління. Основу структурної організації такої пам'яті складають:  $L$  модулів пам'яті, адресний блок і комутуюча мережа. Ємність такої ПП дорівнює:

$$Q = \sum_{i=1}^L Q_i,$$

де  $Q_i$  - ємність  $i$ -го модуля пам'яті. В такій ПП комутаційна мережа реалізується на комутаторах з просторовим розподілом, які забезпечують

з'єднання будь-якого входу з будь-яким виходом або підмножиною виходів. Кожне з'єднання входів і виходів здійснюється за допомогою різної апаратури.. Комутатор з просторовим розподілом реалізується на мультиплексорах, кількість яких дорівнює кількості модулів пам'яті. Для забезпечення з'єднання за повним графом необхідно щоб кожен вхід ПП був з'єднаний з відповідними входами всіх мультиплексорів. Управління комутаційною мережею здійснюється інформацією з виходу БУ.

Апаратні витрати на реалізацію ПП на основі L модулів пам'яті визначаються за формулою:

$$W_1 = LW_{\text{модП}} + W_{\text{БЛА}} + W_{\text{ПК}} + W_{\text{КМ2}},$$

де  $W_{\text{КМ2}}$  – апаратні витрати на реалізацію комутуючої мережі з просторовим розподілом. Час циклу доступу до ПП на основі L модулів пам'яті визначається так:

$$t_{\text{ПП1}} = t_A + t_{\text{ц}} + t_{\text{КМ2}} + t_{\text{н}},$$

де  $t_A$ ,  $t_{\text{ц}}$ ,  $t_{\text{КМ2}}$ ,  $t_{\text{н}}$  – часи відповідно генерації адреси, циклу звертання до модуля пам'яті, затримки даних комутуючою мережею з просторовим розподілом, налаштування комутуючої мережі.

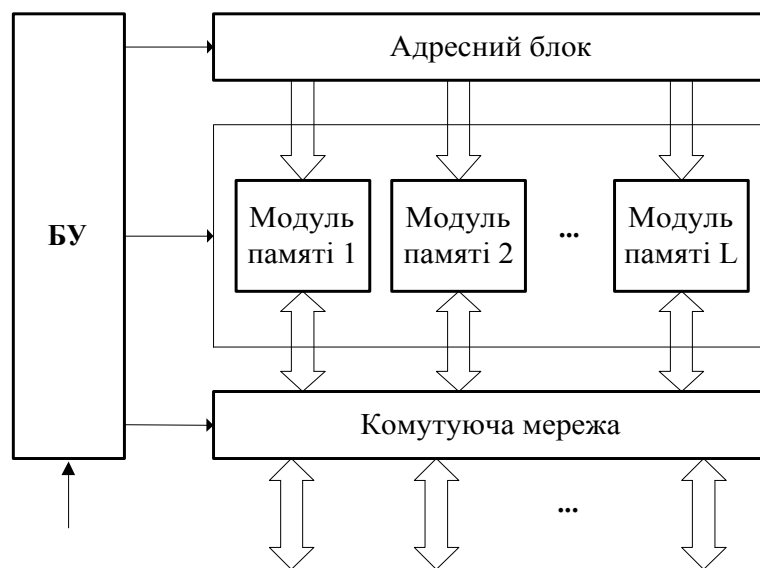


Рисунок 1.5. Базова структура паралельної пам'яті на основі множини модулів пам'яті

Третя базова структура ПП на основі багатоходових комірок пам'яті наведена (рисунок 1.6), де КП - комірка пам'яті, ДШ – дешифратор, КМ – комутуюча мережа. Основу структурної організації такого запам'ятовуючого середовища складає множина комірок пам'яті, доступ до яких є довільним. Особливістю одновимірного багатоходового середовища є можливість одночасного читання інформації з однієї комірки пам'яті  $KP_i$  на декілька виходів [8]. Комутуюча мережа ПП реалізована на базі матричного комутатора, який забезпечує з'єднання  $m$  на  $N$  входів. Особливістю матричного комутатора є те що у разі відсутності конфліктів всі необхідні зв'язки забезпечуються одночасно. Конфлікти виникають тоді, коли два або більше зовнішніх пристроїв вимагають доступу до однієї КП.

Швидкодія такої ПП визначається часом циклу доступу до КП, який дорівнює:

$$t_{цПП_3} = t_A + t_{Дш} + t_{БЕ},$$

де  $t_{Дш}$ ,  $t_{БЕ}$  – часи затримки інформації відповідно на дешифраторі та на буферному елементі. Необхідно відмітити, що ПП на основі такого запам'ятовуючого середовища дозволяє вибирати будь-яку паралельну структуру даних. Основним недоліком ПП на основі багатоходових комірок пам'яті є великі апаратні витрати, що пов'язані з реалізацією буферних елементів, які є основою комутуючої мережі. Апаратні витрати на реалізацію такої ПП визначаються за формулою:

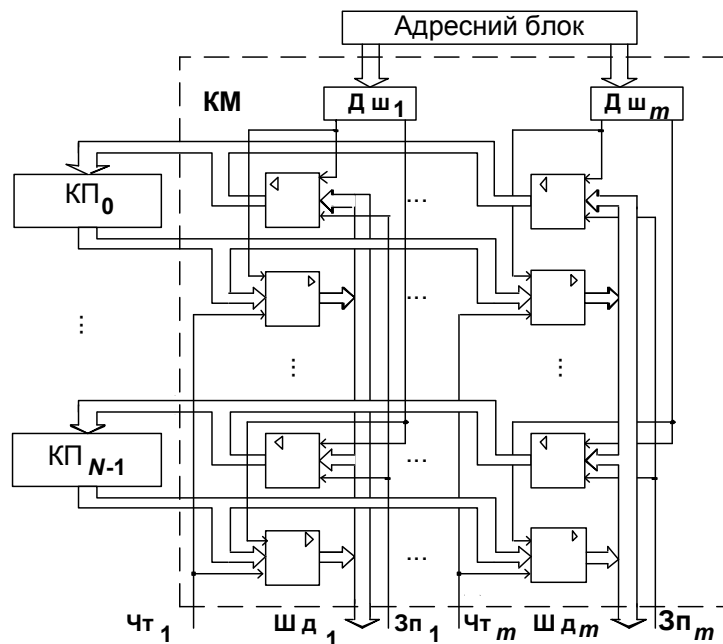


Рисунок 1.6. Базова структура паралельної пам'яті на основі багатовходових комірок пам'яті

$$W_{\text{ППЗ}} = N W_{\text{КП}} + W_{\text{КМЗ}} + W_{\text{БЛА}},$$

де  $W_{\text{КП}}$ ,  $W_{\text{КМ}}$  – витрати обладнання відповідно на комірку пам'яті та матричну комутаційну мережу.

### 1.5 Огляд елементної бази для синтезу запам'ятовуючих середовищ

Найважливіші класифікаційні ознаки запам'ятовуючих пристроїв (ЗП) є :

- спосіб доступу до даних;
- тип робочих режимів (тільки читання або читання і запис);
- схемотехнологія (КМОП, n-МОН, ТТЛ(Ш) і т. д.).

Класифікація за способом доступу до даних і типу робочих режимів приведена на рисунку 1.7. При адресному доступі код на адресному вході вказує осередок, до якого йде звернення. Усі елементи адресної пам'яті

рівнодоступні. Адресні ЗП утворюють найчисленнішу групу мікросхем пам'яті. Інші види пам'яті часто будують на основі адресних ЗП [5].

У ЗП з послідовним доступом записувані дані утворюють деяку чергу. Прочитування походить з черги слово за словом або в порядку запису, або в зворотному порядку. Моделлю такого ЗП являється послідовний ланцюжок осередків, що запам'ятовують, в якому дані передаються між сусідніми осередками.

Прямий порядок прочитування має місце у буферах FIFO з дисципліною "перший прийшов — перший вийшов"(First In — First Out), а також у файлових і циклічних ЗП. Різниця між пам'яттю FIFO і файловим ЗП полягає в тому, що в FIFO запис в порожній буфер відразу ж стає доступним для читання, т. е. слово поступає в кінець ланцюжка. У файлових ЗП дані поступають в початок ланцюжка і з'являються на виході після числа звернень, рівного числу елементів в ланцюжку. При незалежності операцій прочитування і запису фактичне розташування даних в ЗП на момент прочитування не пов'язано з якою-небудь зовнішньою ознакою. Тому записувані дані об'єднують у блоки, що обрамляються спеціальними символами кінця і початку(файли). Прийом даних із ЗП починається після виявлення приймачем символу початку блоку.

Прочитування в зворотному порядку властиве стековим ЗП, для яких реалізується дисципліна "останній прийшов — перший вийшов". Такі ЗП називають буферами LIFO(Last In — First Out).

У циклічних ЗП слова доступні одне за іншим з постійним періодом, визначуваним місткістю пам'яті. До такого типу ЗП відноситься відео пам'ять(VRAM).



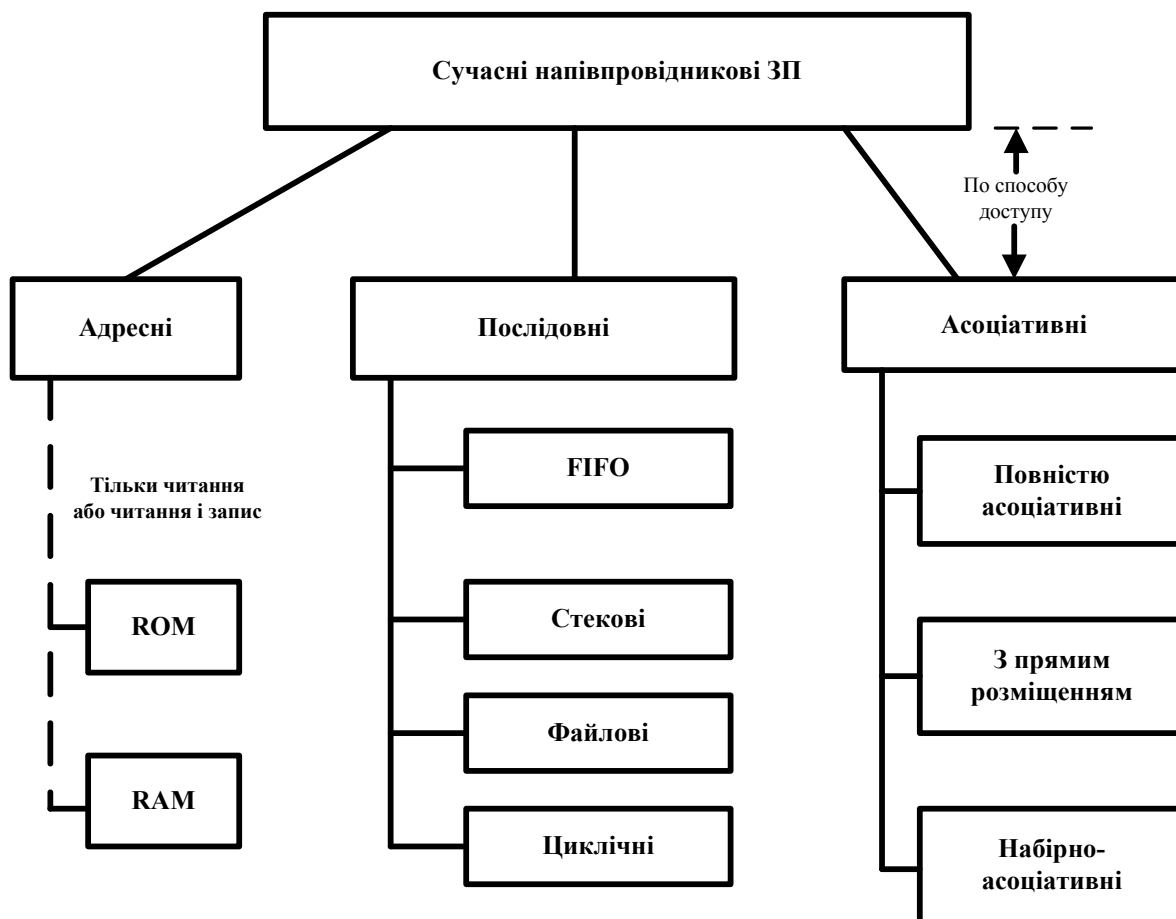


Рисунок 1.7 – Класифікація ЗП за способом доступу до даних і характеру робочих режимів

Час доступу до конкретної одиниці інформації, що зберігається, в послідовних ЗП представляє собою випадкову величину. У найгіршому випадку для такого доступу може знадобитися перегляд усього об'єму даних, що зберігаються.

Асоціативний доступ реалізує пошук інформації за деякою ознакою, а не по її розташуванню в пам'яті (адресі або місцю в черзі). У якнайповнішій версії усі слова, що зберігаються в пам'яті, одночасно перевіряються на відповідність ознаці. На вихід видаються слова, що задовольняють ознаці. Дисципліна видачі слів і запису нових даних можуть бути різними, відповідно до чого розрізняють перераховані в класифікації типи асоціативної пам'яті. Основна сфера застосування асоціативної пам'яті в сучасних ЕОМ — кешування даних [18].

Статичні ОЗП (рисунок 1.8) розділені в класифікації на асинхронні і синхронні. Асинхронні ОЗП названі також стандартними, оскільки раніше вони були практично єдиними представниками статичних мікросхем пам'яті і найбільш звичні для споживача.

Після звернення до асинхронного ЗП до видачі даних проходить час, виявляючим параметром самої пам'яті, не пов'язаним з тактуванням процесора(системи). Відсутність узгодження моментів готовності пам'яті з тактовими сигналами процесора може створити додаткові затримки обміну. Наприклад, можлива видача даних із ЗП зблизька, але після активного фронту тактових імпульсів процесора. У такому разі прийом даних з пам'яті буде відкладений до наступного активного фронту тактового сигналу, т. е. майже на цілий такт.

Асинхронні статичні ОЗП можуть розрізнятися вимогами до сигналів, що управляють: одні не вимагають присутності імпульсних сигналів, інші їх потребують (зазвичай це стосується сигналу CS). У першому випадку відповідний сиг готівка може задаватися як імпульсами, так і рівнями, т. е. бути незмінним і Дозволяючи упродовж багатьох що йдуть підряд циклів звернення до пам'яті. У другому випадку цей сигнал обов'язково має бути імпульсним, наприклад, CS в кожному циклі роботи пам'яті повинен повертатися в пасивне стани і за тим переходити в активне, формуючи перепад в кожному циклі.

У синхронних ОЗП тривалості робочих етапів жорстко пов'язані з тактовими сигналами процесора, і це дозволяє виключити невиправдані втрати часу при обміні, організувати конвеєрну обробку даних, а також позбавити процесорів від простою в очікуванні готовності пам'яті. Таким чином, синхронність пам'яті являється засобом підвищення її швидкодії. Синхронні структури є засобом підвищення її швидкодії. Синхронні структури застосовуються як в статичних, так і в динамічних мікросхемах пам'яті [1].

Серед інших методів підвищення швидкодії статичних ОЗП можна назвати прискорення реверсу шини при переході від передачі даних в одному

напрямку до іншого і використання інтерфейсів DDR, розглянутих далі при описанні динамічних ЗП підвищеної швидкодії.

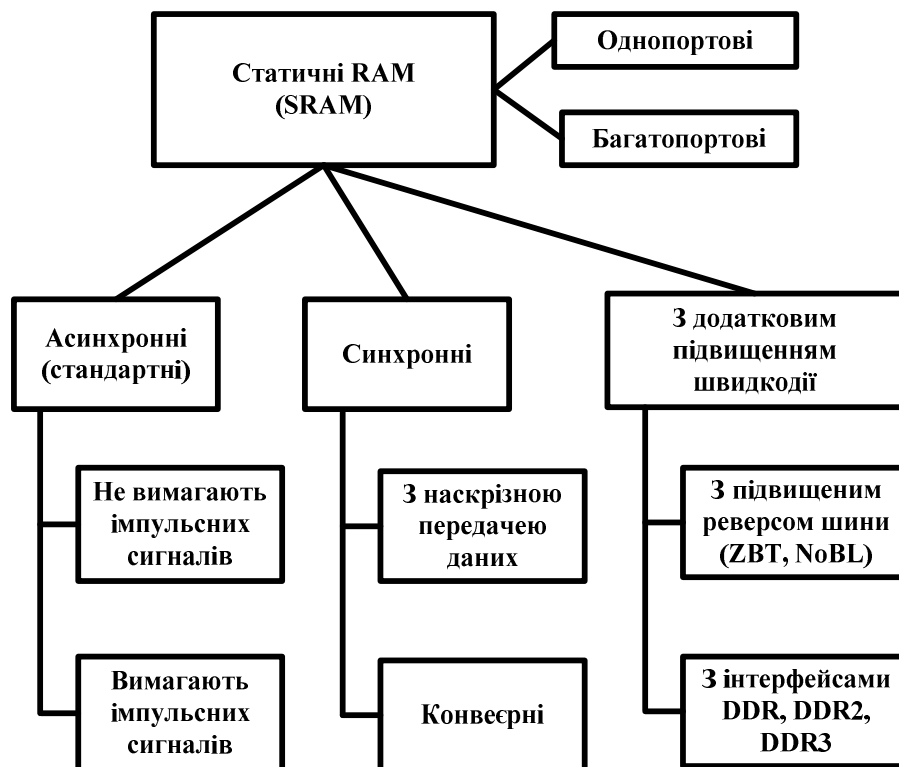


Рисунок 1.8 – Класифікація статичних ОЗП

Статичні ОЗП виконуються як однопортові(звичайні) і багатопортові. У багатопортових ЗП можливі одночасні звернення більш ніж до одного осередку, наприклад, в простих двопортових ЗП можливе прочитування інформації з одного осередку і одночасний запис в іншу, а в істинно двопортових можливе поєднання будь-яких двох операцій. Подібні режими корисні при розділені пам'яті між двома або більше абонентами.

Динамічні ОЗП (рисунок 1.9) характеризуються максимальною інформаційною місткістю і невисокою вартістю, тому саме вони використовуються як основна пам'ять комп'ютерів.

Оскільки бажано отримати від основної пам'яті максимально можлива швидкодія, розроблені численні способи його підвищення. Відповідна архітектура перерахована в класифікації і детальніше розглянута надалі.

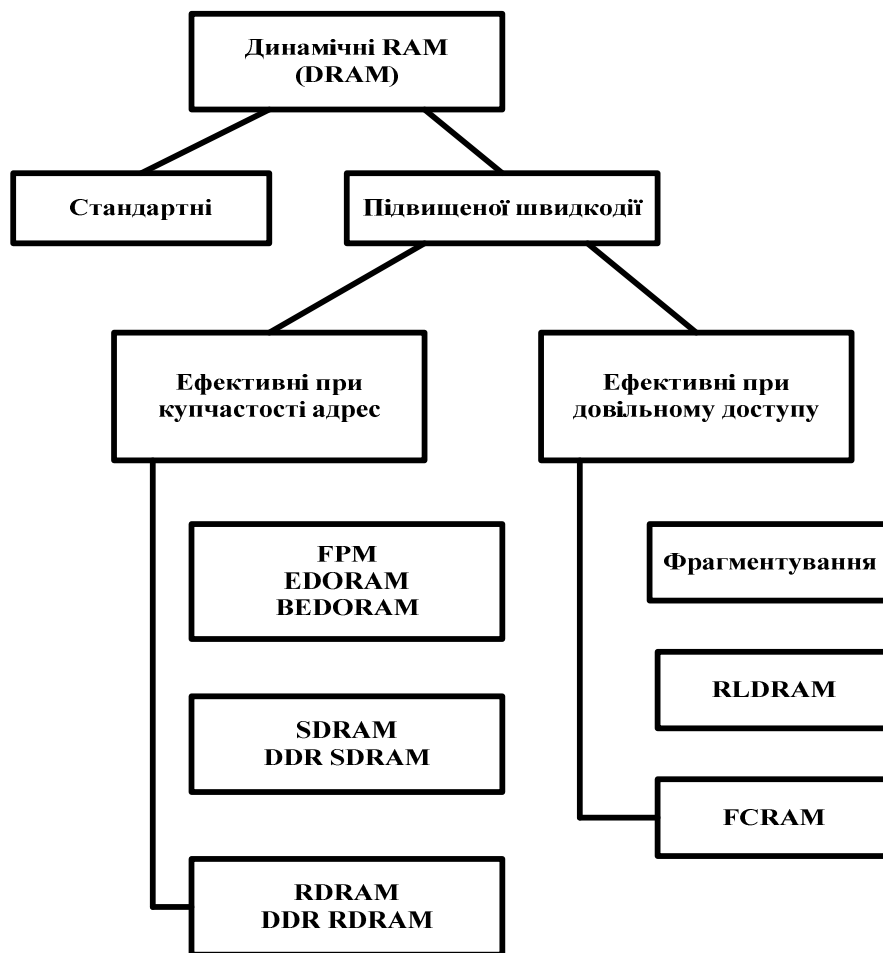


Рисунок. 1.9 – Класифікація динамічних ЗП

Техніко-економічні параметри ЗП істотно залежать від їх схемотехнологічної реалізації. За цією ознакою також можлива класифікація ЗП, однак зручніше розглядати це питання стосовно окремих типів пам'яті.

Багато сучасних засобів обробки інформації вимагають величезних місткостей пам'яті надзвичайно високої швидкодії, що робить актуальним пошук усе більш ефективних рішень. Перспективні варіанти нових типів ЗП перераховані в класифікації на рисунку 1.10.

Найбільш зрілою, такою, що вже досягла рівня промислового виробництва, являється фероелектрична пам'ять (FRAM, Ferroelectric RAM), що поєднує високу ємність і швидкодію з такою корисною властивістю, як незалежність. Таке поєднання властивостей близьке до ідеалу, його не мають ні статичні, ні динамічні ОЗП, ні EEPROM, ні Flash- пам'ять [19].

MRAM(Magneto resistive RAM) — це магніторезистивні ОЗП, в яких елементом, що запам'ятовує, є ділянка магнітного матеріалу, здатна зберігати стан намагніченості за відсутності живлення схеми, так що фізичні властивості використовуваних магнітних матеріалів надають MRAM природну незалежність, а також неруйнівне читання інформації. Міра зрілості MRAM нижча, ніж у FRAM, але їх виробництво вже починається.

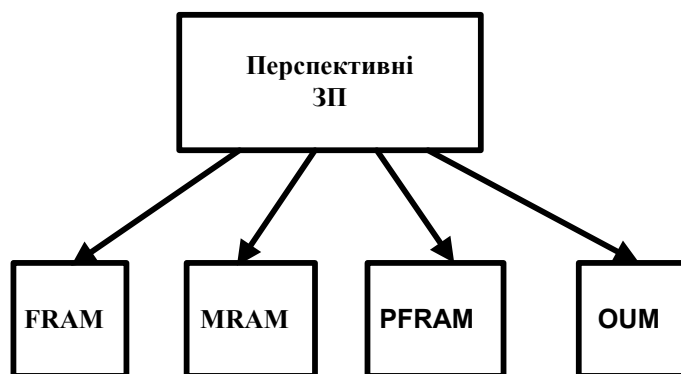


Рисунок. 1.10. Класифікація перспективних ЗП

Полімерні фероелектричні ЗП(PFRAM, Polymeric Ferroelectric RAM), побудовані на основі матеріалів, що утворюють молекулярні ланцюги з диполями. Диполі служать елементами, що запам'ятовують, і залежно від напрямку поляризації зберігають біт інформації. Простота елемента, що запам'ятовує, і компактність конструкції в цілому можуть забезпечити PFRAM надзвичайно високі місткості при дуже малій вартості/біт. В той же час швидкодія PFRAM мало. Це виключає їх застосування в якості ОЗП, але для використання замість дискової пам'яті PFRAM дуже перспективні.

Пам'ять типу OUM(Ovonic Unified Memory) фірми Ovonic реалізована методами інтегральної технології з використанням елементів, що запам'ятовують, властивих компакт-дискам(CD, DVD). Елементом, що запам'ятовує, служить перемичка з халкогенідного сплаву(GeSbTe), який може знаходитися в кристалічному або непровідному аморфному стані, що проводить. Пам'ять вважається дуже перспективною з точки зору економічних показників.

## 1.6 Огляд структур запам'ятовуючих середовищ

Структури адресних ЗП мають багато спільного, що робить раціональним вивчення їх узагальнених структур з подальшим описом елементів, що запам'ятовують, вживаних в різних структурах.

У структурі 2D (рисунок 1.11) елементів ЗЕ, що запам'ятовують, утворюють прямокутну матрицю розмірністю  $M = k * m$ , де  $M$  — інформаційна місткість пам'яті(біт);  $k$ -число слів, що зберігаються;  $m$  — їх розрядність.

Дешифратор адресного коду DC за наявності дозволяючого сигналу CS(Chip Select) активізує відповідно до адреси A одну з вихідних(словникових) ліній, дозволяючи одночасний доступ до усіх елементів вибраного рядка, що зберігає слово, що адресується. Елементи стовпців сполучені вертикальною лінією(розрядною лінією, лінією запису/прочитування) і зберігають однойменні біти усіх слів. Напрямок обміну визначається підсилювачами читання/запису під впливом сигналу R/W(Read /Write) [16].

Структура типу 2D застосовується лише в ЗП малій місткості, т. до. при зростанні місткості різко проявляються її недоліки:

- надмірно складний дешифратор адреси(число його виходів дорівнює числу слів, що зберігаються);
- форма матриці далека від квадратної. Оскільки звичайне число слів, що зберігаються, багаторазово перевищує їх розрядність, число рядків матриці багаторазово перевищує число її стовпців, і матриця набуває вигляду вузької смужки. Реалізація такої матриці ускладнена, і може виявитися абсолютно неприйнятною. Квадратна форма матриці дає великі переваги.

Структура 3D дозволяє різко спростити дешифратори адреси і отримати матрицю квадратної форми за допомогою двокоординатної вибірки елементів, що запам'ятовують. Принцип двокоординатної вибірки пояснимо на прикладі однорозрядної ROM(рисунок 1.12, а). Тут код адреси розрядністю  $p$  ділиться на дві половини, кожна з яких декодується окремо. Вибирається елемент, що

запам'ятовує, розташованих на перетині активних ліній виходів обох дешифраторів. Таких перетинів буде якраз  $2^{n/2} * 2^{n/2} = 2^n$ .

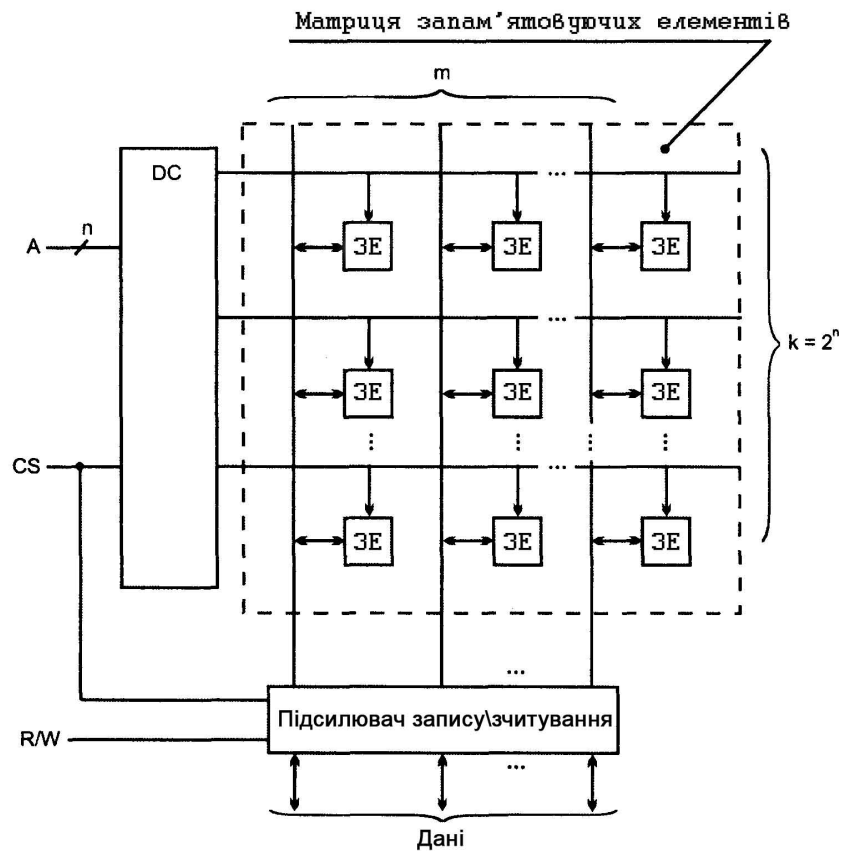


Рисунок 1.11. Структура ЗП типу 3D.

Сумарне число виходів обох дешифраторів складає  $2^{n/2} + 2^{n/2} = 2^{n/2+1}$ , що значно менше ніж  $2^n$ .

Недоліком структури 3D в першу чергу являється ускладнення елементів пам'яті, що мають двокоординатну вибірку. Фактично для вибірки ЗЕ збудженням двох ліній до нього треба додати двовходовий елемент І (рисунок 1.12, б).

Структура 3D може застосовуватися і в ЗП з багаторозрядною організацією (рисунок 1.13), набуваючи при цьому "тривимірного" характеру. В цьому випадку декілька матриць адресуються від одних і тих же дешифраторів, відносно яких вони включені паралельно. Кожна матриця видає один біт адресованого слова, а число матриць дорівнює розрядності слів, що зберігаються.

Структури типу 3D, як і структури 2D, мають обмежене застосування, оскільки сильним конкурентом для них є структура 2DM(2D модифікована).

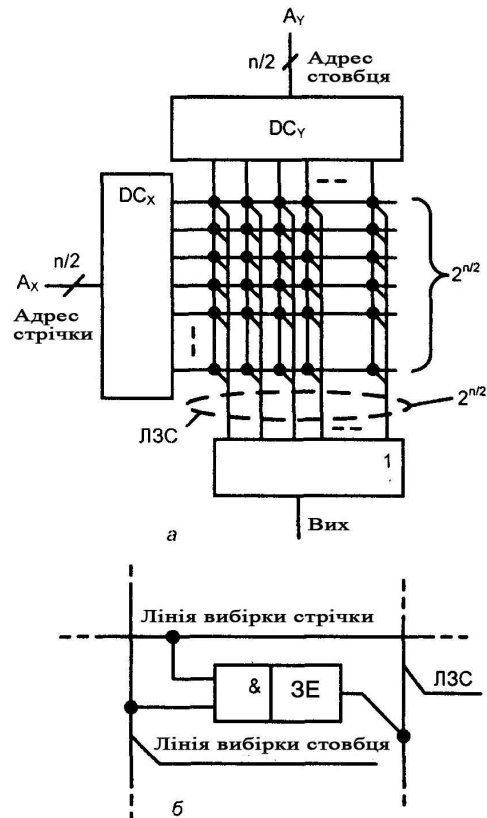


Рис. 2.12. Структура ЗП типу 3D з однорозрядною організацією(а) і елемент(б), що запам'ятовує



Рисунок 2.13 – Структура ЗП типу 3D з багаторозрядною організацією



## **Висновки до розділу**

1. При паралельному методі доступу до окремих блоків пам'яті необхідно забезпечити одночасне встановлення фізичного зв'язку з множиною зовнішніх абонентів і організувати загальне поле пам'яті для даних абонентів.

2. Запам'ятовуюче середовище є основою паралельної пам'яті, воно в значній мірі визначає основні параметри пам'яті: швидкодію, ємність, ширину і варіанти паралельного доступу.

3. Структури адресних ЗП мають багато спільного, що робить раціональним вивчення їх узагальнених структур з подальшим описом елементів, що запам'ятовують, вживаних в різних структурах.

4. Разом з технологічним вдосконаленням елементів, що запам'ятовують, і ланцюгів доступу до них йде і процес модифікації структур ЗП, і те і інше істотно впливає на швидкодію пам'яті, Багато структурних методів набули універсального характеру.

## 2. ПРОЕКТУВАННЯ ЗАСОБІВ ОБМІНУ З БУФЕРИЗАЦІЄЮ ДАНИХ ДЛЯ СИСТЕМ УПРАВЛІННЯ З ПРОМИСЛОВОЮ МЕРЕЖЕЮ

### 2.1. Розробка дерева цілей

Метою даної дипломної роботи є розробка засобів обміну з буферизацією даних для систем управління з промисловою мережею.

Дерево цілей - це графічна схема, яка демонструє розбивку загальних цілей на підцілі. Вершина схеми інтерпретуються як цілі, ребра або дуги - як зв'язки між цілями. Метод дерева цілей є головним універсальним методом системного аналізу. Дерево цілей пов'язує цілі вищого рівня з конкретними засобами їх досягнення на нижчому виробничому рівні через ряд проміжних ланок. Даний метод дозволяє людині привести у порядок власні плани (особисті або професійні), побачити свої цілі в групі.

Щоб розробити ефективну багатопортову пам'ять для комп'ютерних систем реального часу, потрібно зробити структурування мети системи у вигляді дерева цілей, що дозволить отримати відносно стійку структуру цілей, проблем, напрямків, деталізувати цілі і шляхи їхнього досягнення, виявити існуючі між ними взаємозв'язки, забезпечити певну логіку розв'язання проблеми, одержати нові ідеї, розкрити нові можливості розв'язання досліджуваної проблеми [7].

Розроблене дерево цілей наведено на рисунку 2.1. З дерева цілей видно, що для того, щоб розробити ефективну багатопортову пам'ять для комп'ютерних систем реального часу, потрібно вибрати способи доступу до пам'яті, способи запам'ятовуючого середовища, способи звертання до запам'ятовуючого середовища, методи масштабування, а також засоби моделювання.



Рисунок 2.1 – Дерево цілей

## 2.2. Аналіз способу доступу до пам'яті

Для того, щоби отримати можливість використовувати дані з пам'яті в обчислювальних операціях, необхідно однозначно вказати процесору їхнє розташування. В фон-нейманівських машинах кожна комірка пам'яті має власну адресу й проблема визначення місця розташування потрібних даних зводиться до визначення цієї адреси. В перших ЕОМ адреса або номер комірки необхідно було вказувати явно, і такий метод адресації виявлявся дуже незручним. Труднощі в алгоритмізації різних задач, де була потрібна автоматизація процесу визначення адреси, стали причиною введення згодом широкого спектру методів адресації. Кожний з них фактично пропонує певну формулу для обчислення ефективної (тобто фактичної) адреси, зручну в тій або іншій ситуації. Пік винахідництва в цій області припав на час панування CISC-архітектур «регістр-пам'ять», які дозволяли безпосередньо використовувати в якості одного з операндів комірку пам'яті. RISC-архітектури типу «регістр-регістр», в яких доступ до пам'яті регламентується значно більш жорстко, мають у порівнянні з CISC, дуже скромний набір методів адресації, і у найбільш радикальних представниках цієї ідеології зведений до єдиного.

При адресній організації пам'яті розміщення і пошук інформації в масиві, що запам'ятовує, засновані на використанні адреси зберігання слова. Адресою служить номер осередку масиву, в якому це слово розміщується.

При записі (чи прочитуванні) слова в масив, що запам'ятовує, команда, що ініціює цю операцію, повинна вказувати адресу, по якій робиться запис (прочитування) [13].

Типова структура адресної пам'яті містить масив з  $N$   $n$  - розрядних осередків (зазвичай  $n$  дорівнює 1, 4, 8 або 16), що запам'ятовує, і його апаратне обрамлення, що включає регістр адреси  $PtA$ , що має  $k$  розрядів ( $k$  більше або дорівнює логарифму по основі 2 від  $N$ ), інформаційний регістр

РГІ, блок адресної вибірки БАВ, блок підсилювачів прочитування НАМИСТ, блок розрядних підсилювачів-формуваців сигналів запису БУЗ і блок управління пам'яттю БУП.

За кодом адреси в реєстрі адреси блок адресної вибірки формує у відповідному елементі пам'яті сигнали, що дозволяють зробити прочитування або запис слова в осередок.

Цикл звернення до пам'яті ініціюється вступом у блок управління пам'яттю сигналу "Звернення". Загальна частина циклу звернення включає:

- 1) прийом в реєстр адреси з шини адреси ША адреси звернення;
- 2) прийом блоком управління і розшифровку сигналу, що управляє, "Операція", що вказує вид операції (прочитування або запис).

- 3) блок адресної вибірки дешифрує адресу і посилає сигнали прочитування в заданий осередок, при цьому код записаного в осередку слова прочитується підсилювачами прочитування і передається в інформаційний реєстр РГІ;

- 4) у пам'яті з руйнівним прочитуванням (при прочитуванні усі осередки, що запам'ятовують, встановлюються в нульовий стан) робиться регенерація інформації в осередку шляхом запису в неї з інформаційного реєстра РГІ ліченого раніше слова;

- 5) лічене слово видається з інформаційного реєстра на вихідну інформаційну шину ШИВих

- 3) робиться прийом записаного слова з вихідної інформаційної шини ШИВх в інформаційний реєстр;

- 4) блок адресної вибірки робить вибірку і очищення осередку, заданого в реєстрі адреси (у пам'яті з руйнівним прочитуванням для цього робиться прочитування без запису в інформаційний реєстр);

- 5) у вибраній осередок записується слово з інформаційного реєстра.

Блок управління БУП генерує необхідні послідовності сигналів, що управляють, ініціюють роботу окремих вузлів пам'яті.

У пам'яті цього типу пошук потрібної інформації здійснюється не за адресою, а за її змістом (за асоціативною ознакою). При цьому пошук за асоціативною ознакою (чи послідовно по окремих його розрядах) відбувається паралельно в часі для усіх осередків масиву, що запам'ятовує. У багатьох випадках асоціативний пошук дозволяє істотно спростити і прискорити обробку даних. Це досягається за рахунок того, що в пам'яті цього типу операція прочитування інформації поєднана з виконанням ряду логічних операцій.

Пам'ять цього типу застосовується в спеціалізованих обчислювальних машинах - машинах баз даних.

Буфер представляє собою ЗП для зберігання черг даних(списків) з порядком вибірки слів, таким же, що і порядок їх вступу. Моменти запису слова у буфер і прочитування з нього задаються зовнішніми сигналами управління незалежно один від одного.

Різний темп прийому і видачі слів потрібний, наприклад, якщо приймач здатний приймати дані, що поступають регулярно з деякою частотою, а джерело видає слова в швидшому темпі і, можливо, до того ж не регулярно. Такі дані поступають в їх темпі у буфер FIFO, а з нього прочитуються регулярно з частотою, необхідною для приймача. Нове слово ставиться в кінець черги, прочитування здійснюється з початку черги [16].

Перед початком роботи обидва лічильники адрес CTR 1 і CTR 2 скидаються. При записі адреси збільшуються на одиницю при кожному зверненні, т. е. зростають, починаючи з нульового. Те ж відбувається при читанні слів, так що адреса читання завжди "женеться" за адресою запису. Якщо адреси порівнюються при читанні, то буфер порожній. Якщо адреси порівнюються при записі, то буфер повний(адресами зайнята уся місткість лічильника). Ці ситуації відзначаються відповідними сигналами. Якщо буфер повний, то треба припинити прийом даних, а якщо порожній, то треба припинити читання. Черга подовжується або коротшає залежно від різниці чисел записаних і лічених слів. Перехід через нуль ускладнень не викликає.

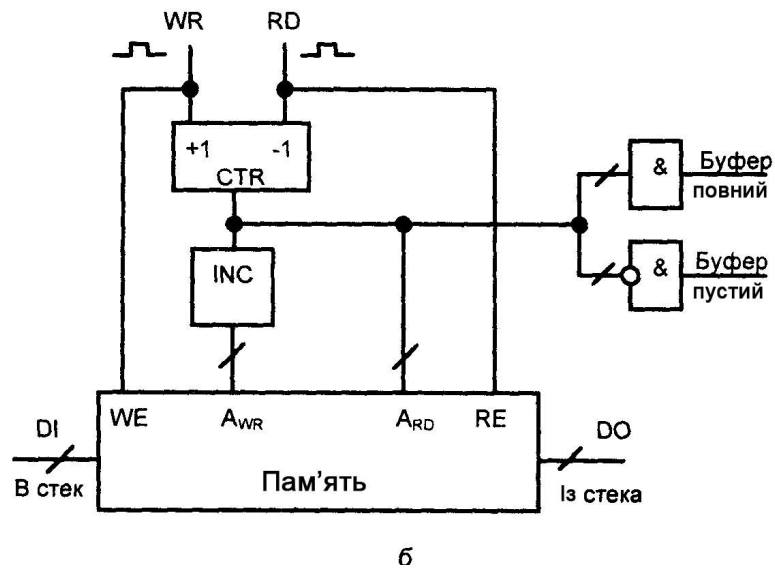
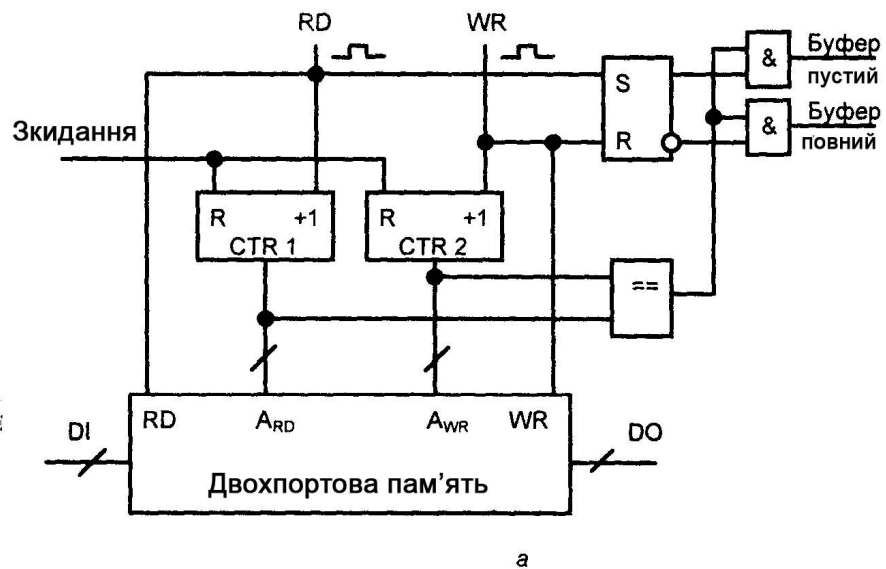


Рисунок 2.2 - Структури буферів FIFO(а) і LIFO(б)

Це ЗП для зберігання черги даних з порядком прочитування, зворотним порядком запису(останній прийшов — перший вийшов). Приклад застосування зберігання станів реєстрів комп'ютера при перериванні основної програми і переході до іншої і подальше повернення даних в ті ж реєстри при відновленні виконання основної програми.

Приклад буфера LIFO наведений на рисунок 2.2, б. Адреси слів при зверненні до буфера формуються в реверсивному лічильнику. При записі адреса збільшується, а при читанні зменшується на одиницю. Схема INC(Incrementation) збільшує передаване через неї число на одиницю. Після записів в лічильнику знаходиться деяке число. Читання відбуватиметься за тією

адресою, яка знаходиться в лічильнику, а запис по сусідньому більшому. Звідси видно, що стан лічильника вказує вершину стека — останній зайнятий осередок блоку пам'яті. Стани "буфер повний" і "буфер порожній" виявляються аналізом утримуваного лічильника. Код, що складається з одиниць, вказує на те, що буфер повністю заповнений, а код, що складається з нулів, — на те, що буфер порожній. Індикація цих кодів виконується кон'юнкторами з прямими і інверсними входами [5].

Інший варіант реалізації буфера LIFO — набір реверсивних регістрів, число яких дорівнює розрядності слів. При одному напрямі зрушення дані всовуються в регістри(записуються), при іншому висуюються(читаються). Першим висувається останнє записане слово.

### 3.3. Аналіз способу побудови запам'ятовуючого середовища

Існуючі структури паралельної пам'яті комп'ютерних систем реального часу відрізняються як за способами побудови запам'ятовуючого середовища, структурною організацією комутуючої мережі, розміщенням і пошуком інформації, так і за способом узгодження інтенсивності надходження даних із пропускною здатністю паралельної пам'яті [18]. Аналіз структур паралельної пам'яті дозволив виділити три базові структури:

- на основі багатовимірного запам'ятовуючого середовища;
- на основі одновимірного багатовходового середовища.

Аналіз багатовимірного запам'ятовуючого середовища. Основу структурної організації такої пам'яті становить багатовимірне запам'ятовуюче середовище та  $m$ -входова комутуюча мережа. Базова структура паралельної пам'яті на основі багатовимірного запам'ятовуючого середовища наведена на рисунку 2.3, де ГА - генератор адреси, ОЗП - оперативний запам'ятовуючий пристрій.



Багатовимірне запам'ятовуюче середовище паралельної пам'яті реалізується на базі на множини модулів. Загальну ємність запам'ятовуючого середовища визначають за формулою:

$$Q_{\text{III}} = \sum_{i=1}^m Q_i$$

де  $Q_i$  - ємність  $i$ -го модуля пам'яті. В такій паралельній пам'яті комутаційна мережа залежно від кількості модулів пам'яті  $t$  може реалізовуватися у вигляді просторової, матричної або складеної багатоярусної комутуючої мережі. Блок управління налаштовує комутуючу мережу на необхідні з'єднання та ГА - на генерацію послідовностей адрес відповідно до вимог алгоритмів обробки [9].

Апаратні затрати на реалізацію паралельної пам'яті на основі багатовимірного запам'ятовуючого середовища переважно залежать від двох параметрів: ширини доступу та ємності модулів пам'яті. Необхідні апаратні затрати на реалізацію ПП визначаються за формулою:

$$W_{\text{ПП}_2} = mW_{\text{модП}} + W_{\text{ЕлА}} + W_{\text{ПК}} + W_{\text{КМ}_2}$$

де  $W_{\text{КМ}_2}$  - апаратні витрати на реалізацію комутуючої мережі;  $W_{\text{модП}}$  - апаратні витрати на реалізацію модуля пам'яті. Час циклу доступу до паралельної пам'яті на основі  $t$  модулів пам'яті дорівнює:

$$t_{\text{ПП}_1} = t_{\text{ГА}} + t_{\text{ц}} + t_{\text{КМ}_2}$$

де  $t_{\text{ГА}}$ ,  $t_{\text{ц}}$ ,  $t_{\text{КМ}_2}$ ,  $t_{\text{н}}$  - тривалості відповідно генерації адреси, звертання до модуля пам'яті та затримки даних комутуючою мережею

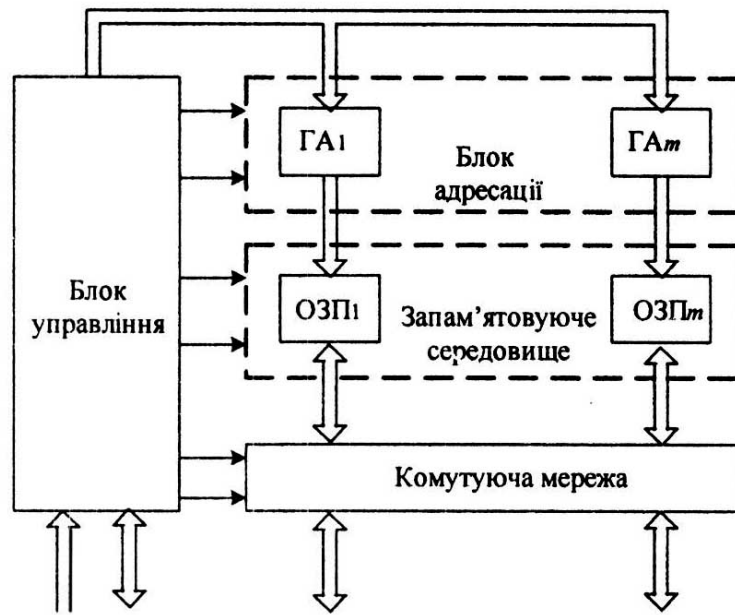


Рисунок 2.3 - Базова структура паралельної пам'яті на основі багатовимірного запам'ятовуючого середовища.

Структурна організація такої пам'яті будується на одновимірному запам'ятовуючому середовищі на основі багатовходових комірок пам'яті та матричній комутуючій мережі.

Базова структура паралельної пам'яті на основі одновимірного багатовходового середовища наведена на рис. 3.4, де КП - комірка пам'яті, Дш - дешифратор

Одновимірне запам'ятовуюче середовище на основі численних багатовходових комірок пам'яті уможливорює одночасне читання інформації з однієї комірки пам'яті КП на декілька виходів.

Доступ до комірок пам'яті запам'ятовуючого середовища забезпечується матричною комутуючою мережею розміром  $m \times N$ , яка реалізується на буферних елементах. Особливістю матричної комутуючої мережі є те, що у разі відсутності конфліктів всі необхідні зв'язки виконуються одночасно. Конфлікти виникають тільки тоді, коли два або більше зовнішніх пристроїв хочуть здійснити запис в одну і ту КП [3].

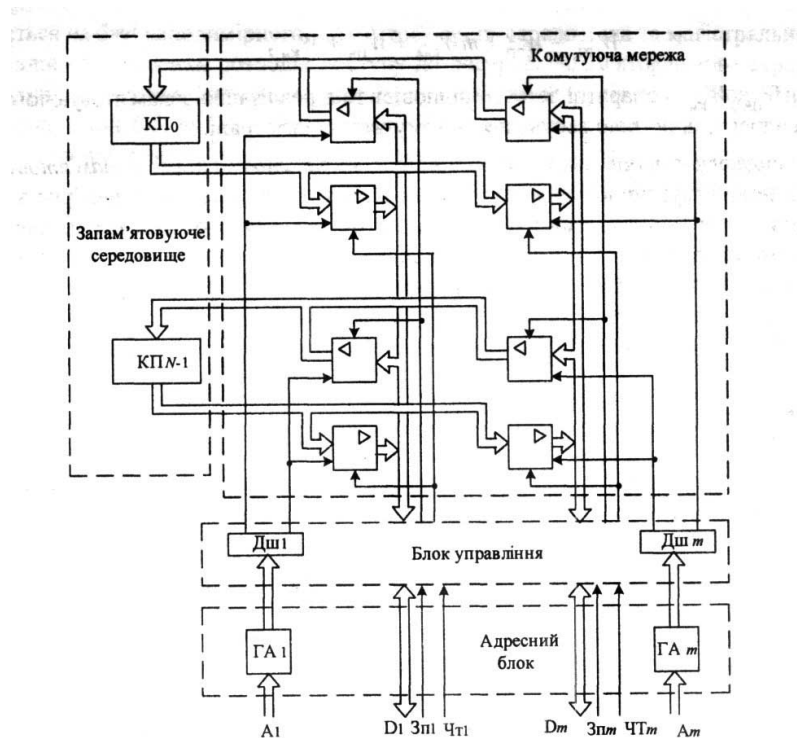


Рисунок 2.4 - Базова структура паралельної пам'яті на основі одновимірного багатовходового середовища

Основним недоліком паралельної пам'яті на основі одновимірного багатовходового середовища є великі апаратні затрати, що пов'язані з реалізацією матричної комутуючої мережі. Апаратні затрати на реалізацію такої ПП визначають за формулою:

$$W_{ПП_2} = NW_{КП} + W_{КМ_2} + W_{БЛА}$$

де  $W_{КП}, W_{КМ}$  - витрати обладнання відповідно на комірку пам'яті та матричну комутаційну мережу.

Швидкодія такої паралельної пам'яті зумовлена часом циклу доступу до КП, який дорівнює:

$$t_{цПП_2} = t_{ГА} + t_{дш} + t_{БЕ}$$

де  $t_{дш}, t_{БЕ}$  - тривалість затримки інформації відповідно на дешифраторі та на буферному елементі. Необхідно відмітити, що паралельна пам'ять на основі

такого запам'ятовуючого середовища дозволяє вибирати будь-яку паралельну структуру даних.

#### 2.4. Аналіз способу звертання до запам'ятовуючого середовища

У комп'ютерних системах реального часу для досягнення високої ефективності використання обладнання вимагається узгодження інтенсивності надходження даних з інтенсивністю доступу до паралельної пам'яті. Для реалізації такого узгодження використовуються методи і засоби послідовно-паралельного, паралельно-послідовного перетворень та їх комбінацій [10]. Під час реалізації перетворень необхідно виконання такої умови:

$$m_d n_d F_d \leq m_{ПП} n_{ЗС} F_{ПП}, \quad (2.1)$$

де  $m_d$  - кількість каналів надходження даних,  $n_d$  - розрядність даних,  $F_d$  - частота надходження даних,  $m_{ПП}$  - ширина доступу до паралельної пам'яті,  $n_{ЗС}$  - розрядність запам'ятовуючого середовища,  $F_{ПП}$  - частота доступу до паралельної пам'яті. З формули (2.1) видно, що інтенсивність доступу до паралельної пам'яті залежить як від ширини і частоти доступу, так і розрядності запам'ятовуючого середовища.

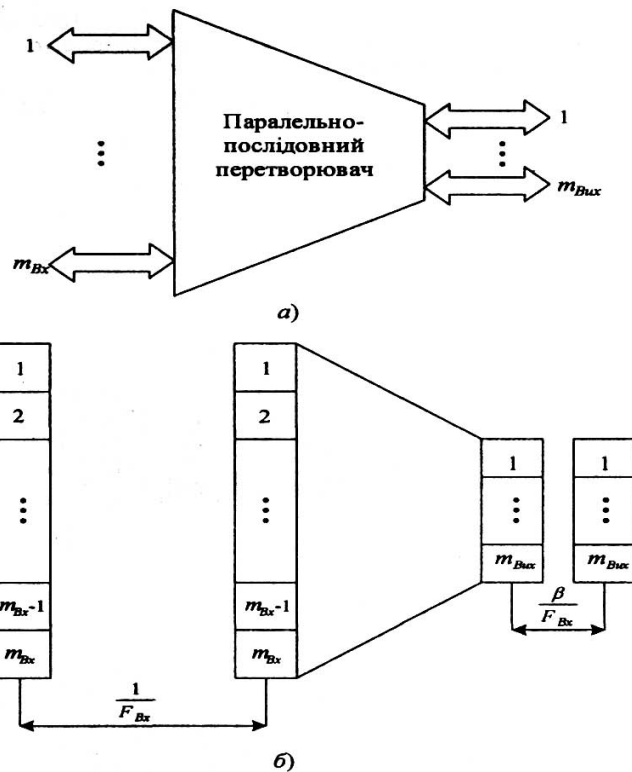


Рисунок 2.5 - Паралельно-послідовне перетворення: а) ширини доступу; б) частоти доступу

За допомогою послідовно-паралельного або паралельно-послідовного перетворення можна змінювати ширину доступу та частоту звертання до пам'яті. Реалізація таких перетворень ґрунтується на виконанні рівності:

$$m_{Bx} n_{Bx} F_{Bx} = m_{Bix} n_{Bix} F_{Bix} \quad (2.2)$$

де  $n_{Bx}$  - кількість каналів на вході перетворювача,  $n_{Bix}$  - розрядність даних на вході перетворювача,  $F_{Bx}$  - частота надходження даних на вхід перетворювача,  $n_{Bix}$  - кількість каналів на виході перетворювача,  $n_{Bix}$  - розрядність даних на виході перетворювача,  $F_{Bix}$  - частота даних на виході перетворювача. Кожне перетворення характеризується коефіцієнтом, який визначають так:

$$\beta = \frac{n_{Bx} n_{Bx}}{n_{Bix} n_{Bix}} \quad (2.3)$$

Паралельно-послідовне перетворення дає змогу зменшити кількість виходів перетворювача (ширини доступу до пам'яті  $m_{\text{Вих}}$ ) та збільшити частоту даних на виході перетворювача ( $F_{\text{Вих}}$  - частота доступу до пам'яті).

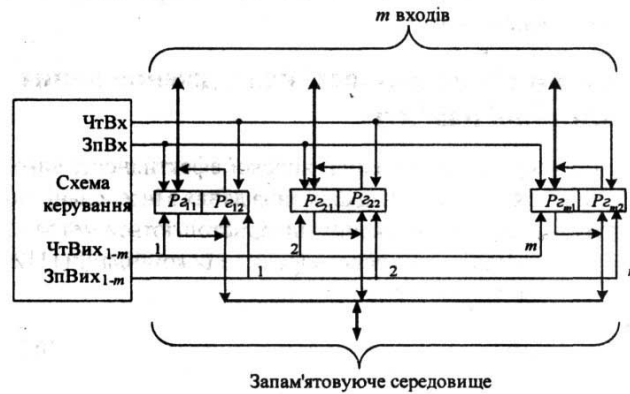


Рисунок 2.6 - Структурна схема пристрою паралельно-послідовного перетворення

Для узгодження інтенсивностей надходження даних і доступу до пам'яті кількість виходів  $t_{\text{Вих}}$  і частоту на виході  $F_{\text{Вих}}$  паралельно-послідовного перетворювача обчислюють за виразами:

$$m_{\text{Вих}} \geq \beta m_{\text{Вх}}; F_{\text{Вих}} \geq \frac{F_{\text{Вх}}}{\beta}$$

Схема, яка ілюструє паралельно-послідовне перетворення ширини і частоти доступу до запам'ятовуючого середовища, наведена на рисунку 2.5.

Паралельно-послідовне перетворення використовується при побудові паралельної пам'яті на базі одновимірного запам'ятовуючого середовища. Параметри запам'ятовуючого середовища (цикл доступу  $t_{\text{цзс}}$  і розрядність  $n$ ) є визначальними для вибору коефіцієнта  $P$  перетворення. У паралельно-послідовному перетворювачі відбувається масштабування часу відповідно до рисунка 2.5б [9].

Паралельно-послідовне перетворення реалізує пристрій, структура якого наведена на рисунку 2.6.

Основними компонентами такого пристрою перетворення є схема управління та  $2t$  регістрів з тристабільними виходами. Особливістю паралельно-послідовного перетворювача (рисунок 2.6) є зменшення ширини доступу з  $t$  входів до одного та забезпечення двонапрявленого передавання даних.

У такому пристрої при записі вхідних даних у запам'ятовуюче середовище вони спочатку сигналом  $Z_{пВх}$  паралельно записуються у регістри  $R_{Г11} - R_{Гm1}$ , а потім сигналами  $Ч_{ТВих 1} \dots, Ч_{ТВих m}$  послідовно зчитуються на шину запам'ятовуючого середовища. При читанні даних із запам'ятовуючого середовища вони спочатку сигналами  $Z_{пВих 1} \dots, Z_{пВих m}$  послідовно записуються у  $R_{Г12} - R_{Гm2}$ , а потім сигналом  $Ч_{ТВх}$  паралельно зчитуються на вхідні шини перетворювача. У схемі управління формування сигналів  $Z_{пВх}$  і  $Ч_{ТВх}$  здійснюється з частотою  $F_{Вх}$ , а сигналів  $Ч_{ТВих1}, \dots, Ч_{ТВихm}$  та  $Z_{пВих 1} \dots, Z_{пВих m}$  з частотою  $F_{Вх}\beta$ . Для реалізації паралельно-послідовного пристрою перетворення необхідні такі апаратні затрати:

$$W_1 = W_{СУ} + 2m W_{РГ}$$

де  $W_{СУ}$ ,  $W_{РГ}$  - апаратні витрати на реалізацію відповідно схеми управління та регістрів.

Аналіз паралельно-послідовного перетворення. Послідовно-паралельне перетворення збільшує ширину та зменшує частоту доступу до запам'ятовуючого середовища. Особливістю такого перетворення є те, що коефіцієнт перетворення  $\beta$  є завжди більшим за одиницю. Частота доступу на коефіцієнт  $\rho$  до запам'ятовуючого середовища зменшується за рахунок збільшення у  $\beta$  разів його розрядності. Використання послідовно-паралельного перетворення дозволяє здійснювати у реальному часі доступ до паралельної пам'яті у пристроїв з швидкодією у  $\beta$  разів більшою, ніж швидкодія запам'ятовуючого середовища. Схема, яка ілюструє послідовно-паралельного перетворення ширини і частоти доступу, наведена на рисунку 2.7 [3].

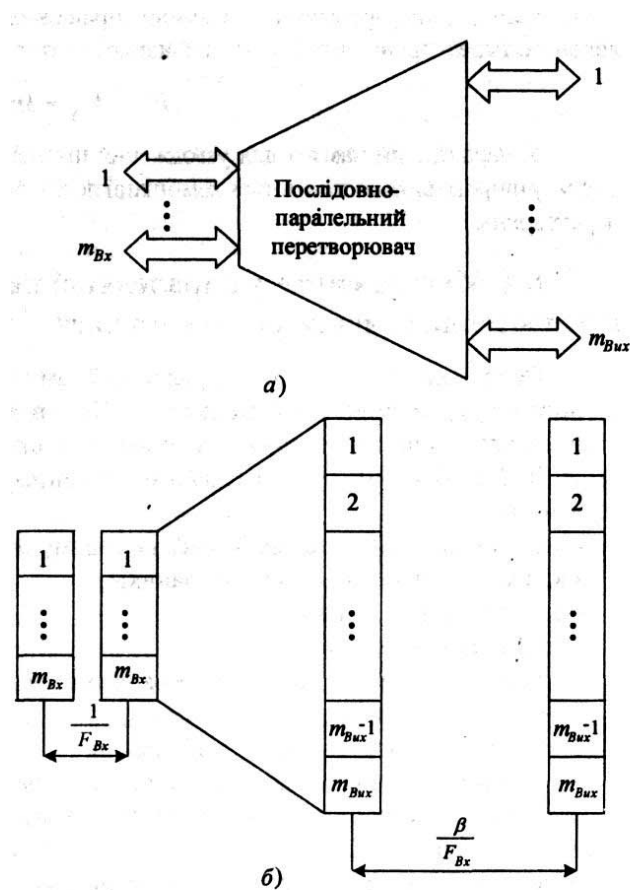


Рисунок 2.7. - Послідовно-паралельне перетворення: а) ширини доступу; б) частоти доступу

Послідовно-паралельне перетворення реалізує пристрій, структура якого наведена на рисунку 2.8.

Структура пристрою послідовно-паралельного перетворення є складнішою від паралельно-послідовного, оскільки в його склад додатково введено  $m$  буферних регістрів БРГ, призначених для проміжного зберігання вхідних даних. Для забезпечення двонапрявленого передавання даних у регістрах  $РГ_{11}, \dots, РГ_{1m}$  і  $РГ_{21}, \dots, РГ_{2m}$  виходи є тристабільними. Запис вхідних даних у буферні регістри  $БРГ_1 \dots, БРГ_m$  здійснюється послідовно з частотою  $F_{Bx}$ , а регістри  $РГ_{11}, \dots, РГ_{1m}$  і  $РГ_{21}, \dots, РГ_{2m}$  запам'ятовуюче середовище - паралельно з частотою  $\beta F_{Bx}$ . При читанні даних із запам'ятовуючого середовища вони спочатку сигналом  $Zп^{\wedge}$  паралельно записуються у регістри  $РГ_{12} - РГ_{m2}$ , а потім сигналами  $ЧГ_1 \dots, ЧГ_m$  послідовно з частотою  $F_{Bx}$  зчитуються на вхідну шину перетворювача. Сигнали запису та читання  $РГ_{11}, \dots, РГ_{1m}, РГ_{21}, \dots, РГ_{2m}$  і  $БРГ_1, \dots, БРГ_m$  формуються схемою управління.



Для реалізації пристрою послідовно-паралельного перетворення необхідні такі апаратні затрати:

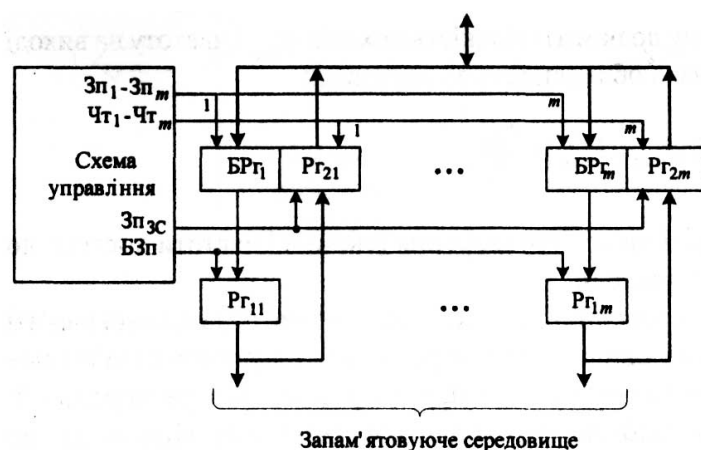


Рисунок 2.8 - Структурна схема пристрою послідовно-паралельного перетворення

## 2.5. Аналіз комутуючої мережі

Комутуюча мережа паралельної пам'яті забезпечує необхідне з'єднання між запам'ятовуючим середовищем та виходами пам'яті. Характеристики комутуючої мережі (КМ) значною мірою впливають на характеристики систем паралельної пам'яті. Однією із основних характеристик КМ є пропускна здатність і можливості з'єднання, які для одних КМ обмежуються кількома варіантами комутації входів з виходами, а в інших - забезпечують довільні варіанти з'єднань для передавання даних. Пропускна здатність КМ, переважно, зумовлена: числом входів і виходів;

структурною організацією; алгоритмом арбітражу; типом управління; розрядністю шин передавання даних; способом синхронізації передачі даних.

Основою КМ є комутатори, які можна поділити на три класи - прості, матричні та складені, які реалізуються на базі простих. Прості комутатори характеризуються малим часом з'єднання та невеликою кількістю входів та

виходів. У разі необхідності з'єднання великої кількості входів з виходами використовуються складені комутатори, які реалізуються з простих шляхом їх об'єднання в багатоярусні схеми. Прості комутатори поділяють на два види: з часовим і просторовим розділенням доступу. Особливістю таких комутаторів є простота управління та висока швидкодія.

Комутатори з часовим розподілом доступу до магістралі запам'ятовуючого середовища називаються магістральними. Особливістю таких комутаторів є використання загальної інформаційної магістралі для передавання даних між запам'ятовуючим середовищем та пристроями, які підключаються до магістралі. Переважно, шини в таких структурах реалізуються з пасивних елементів і все управління обміном виконується контролерами пам'яті. Інколи між пристроями, які підключені до шини, виникають конфлікти з приводу використання спільного ресурсу, тому необхідні засоби управління доступом до шини [20]. Для розв'язання конфліктів у разі одночасного запиту на доступ до шини декількох пристроїв використовують такі методи:

- призначення кожному пристрою унікального пріоритету;
- обслуговування за дисципліною FIFO (перший прийшов, перший вийшов);
- з'єднання пристроїв в ланцюжок або коло з фіксацією їх пріоритетності.

Здебільшого для доступу до магістралі паралельної пам'яті використовується алгоритм фіксованих часових інтервалів [10].

Хоча комутатор з часовим розподілом вимагає невеликих апаратних затрат, він має численні обмеження. Основними серед них є збільшення часу арбітражу та часу доступу до шини у разі збільшення числа пристроїв, що підключаються.

Комутатори з просторовим розподілом доступу до запам'ятовуючого середовища з'єднують будь-який вхід запам'ятовуючого середовища з будь-яким входом або підмножиною входів комутатора. Кожне з'єднання входів і виходів здійснюється за допомогою різної апаратури. Структура таких

комутаторів наведена на рисунку 2.9, де MUX - мультиплексор, РгБ і РгУ - регістри відповідно буферний і управління [10].

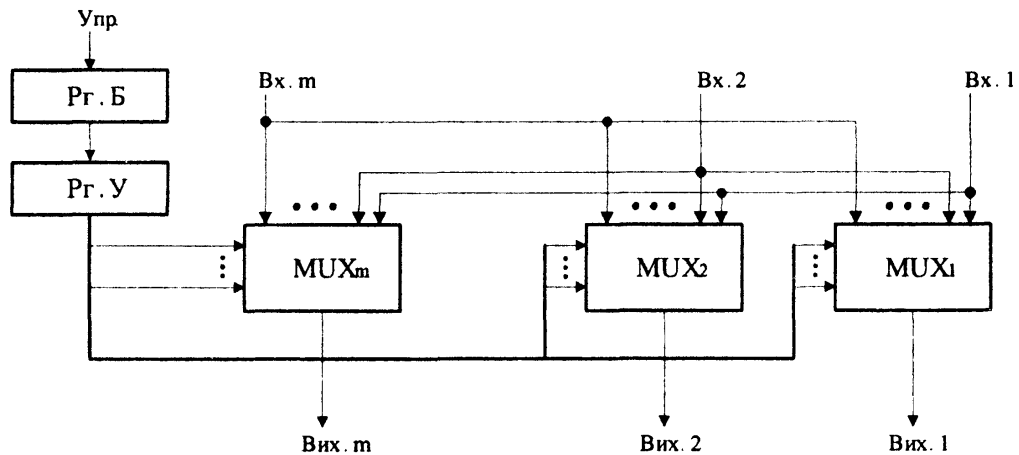


Рисунок 2.9. Структура просторового комутатора

Комутатор з просторовим розподілом реалізується на мультиплексорах, кількість яких дорівнює кількості виходів комутатора. Кількість входів комутатора може бути довільною. Для забезпечення з'єднання за повним графом необхідно, щоб кожний вхід комутатора був з'єднаний з відповідними входами всіх мультиплексорів. Комутатором управляє інформація з виходу регістра РгУ. Введення до складу комутатора буферного регістра РгБ дозволяє сумістити в часі процес програмування нового порядку комутації з передаванням даних [19].

Основні переваги такого комутатора - повна доступність і мінімальна затримка передавання даних, а головні недоліки - великі витрати обладнання.

Матричні комутатори є найгнучкішим типом КМ. Структура матричного комутатора, який з'єднує  $M$  на  $N$  входів, наведена на рисунку 2.10.

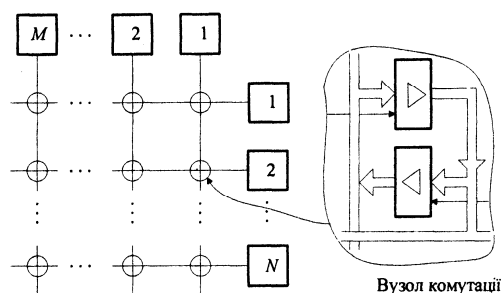


Рисунок 2.10 - Матричний комутатор  $M \times N$

Особливістю матричного комутатора є те, що у разі відсутності конфліктів всі необхідні зв'язки забезпечуються одночасно. Конфлікти виникають тоді, коли два або більше пристроїв вимагають комутації одного і того ж вузла. Число вузлів комутації дорівнює добутку  $M \times N$ . Матричні комутатори доцільно використовувати за невеликої кількості входів і виходів.

Нескладні просторові та матричні комутатори з великою кількістю входів і виходів вимагають великих витрат обладнання. Неefективними в цьому випадку є також прості комутатори з часовим розподілом, які мають невисоку швидкодію [16].

Складені багатоярусні комутуючі мережі синтезуються на базі комутуючих елементів (КЕ). Як правило, КЕ - це комутатори з двома входами та двома виходами. Складені багатоярусні КМ можна поділити на три основні типи:

- з блокуванням, які можуть мати стан, коли неможливо знайти шлях з'єднання незайнятого вхідного порту з незайнятим вихідним портом;
- неблоковані, в яких завжди можна знайти шлях з'єднання вільної пари портів;
- з переналаштуванням, які виконують всі можливі з'єднання між вхідними і вихідними портами переналаштування існуючих зв'язків.

Динамічні переставляння виконуються в КЕ, які переважно є однонапрям-леними. Для побудови двонапрямлених КМ розроблено КЕ, структура якого наведена на рисунку 2.11.

Двонапрямлений КЕ складається з двох мультиплексорів 4:1, з незалежним керуванням та чотирьох шинних формувачів. Такий КЕ може перебувати у восьми різних станах які сигналами  $S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7$

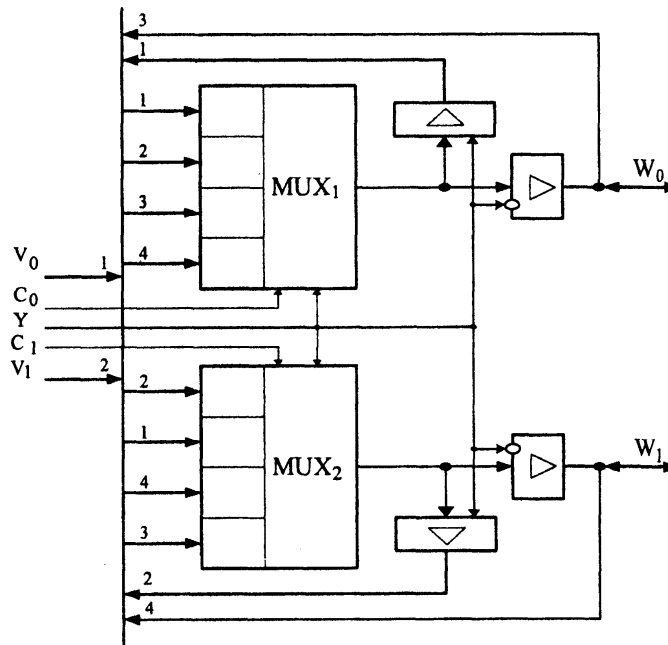


Рисунок 2.11 - Схема двонапрявленого комутуючого елементу.

## 2.6. Аналіз методів управління доступом до паралельної пам'яті

У паралельній пам'яті можливі декілька випадків, коли пристрої можуть конфліктувати при доступі до інформації [15]. Переважно ці конфлікти виникають в ситуаціях, коли два порти намагаються одночасно отримати доступ до однієї комірки пам'яті. Можна виділити два випадки: перший, коли один порт записує дані, а інший зчитує інформацію; другий, коли обидва порти записують інформацію одночасно за тією ж адресою. Важливо також забезпечити пріоритетність процедури запису над процедурою читання, щоб у разі одночасного доступу до однієї комірки пам'яті зчитана інформація не була застарілою.

Перший випадок носить технічну назву “Flow thru” і описує ситуацію, коли дані, які щойно були записані одним портом, відразу зчитуються іншим портом, немов “пропливаючи” через елемент пам'яті. Це вимагає, щоб один пристрій мав вищий пріоритет (master device) відносно до іншого (slave device).



Проаналізуємо роботу даної схеми для трьох можливих варіантів доступу до пам'яті:

- адреси звертань портів R і L не однакові;
- правий порт звертається за тією ж адресою, якою вже володіє лівий порт;
- правий і лівий порти звертається за однією адресою одночасно.

Якщо адреси звертань портів R та L не однакові, то сигнали ЗАЙНЯТО та WR\_ЗАБОРОНА приймають високий логічний рівень і процедура арбітражу портів не проводиться.

Роботу схеми для другого варіанту доступу пояснює діаграма, приведена на рис. 3.15, де -  $t_{BLA}$  - максимальний час переходу сигналу ЗАЙНЯТО від моменту співпадіння адрес на портах до рівня лог. 0,  $t_{BHA}$  - максимальний час переходу сигналу ЗАЙНЯТО від моменту неспівпадіння адрес на портах до рівня лог. 1,  $t_{BDD}$  - максимальний час коли дані стають доступними для порту, після того як зникає сигнал ЗАЙНЯТО,  $t_{DDD}$  - максимальний час між початком запису нової інформації лівим портом в пам'ять і моментом, коли цими даними може розпоряджатись правий порт,  $t_{WDD}$  - максимальний час між початком сигналу WE\_L (дозвіл запису лівого порту) і моментом, коли даними може розпоряджатись правий порт [13].

На рис.3.13 показана ситуація коли лівий порт першим заволодів правом доступу до комірки пам'яті для процедури запису (сигнал WE\_L) даних DATA\_IN\_L. Для правильної роботи схеми необхідно, щоб час затримки D був більший суми часів спрацювання компаратора, 2-х логічних елементів "І" і зовнішнього пристрою обробки сигналів ЗАЙНЯТО\_R та WR\_ЗАБОРОНА.

В третьому варіанті доступу на виходах обох компараторів встановились рівні лог. 1, арбітражне блокування порту визначається випадковим станом виходів RS-тригера.

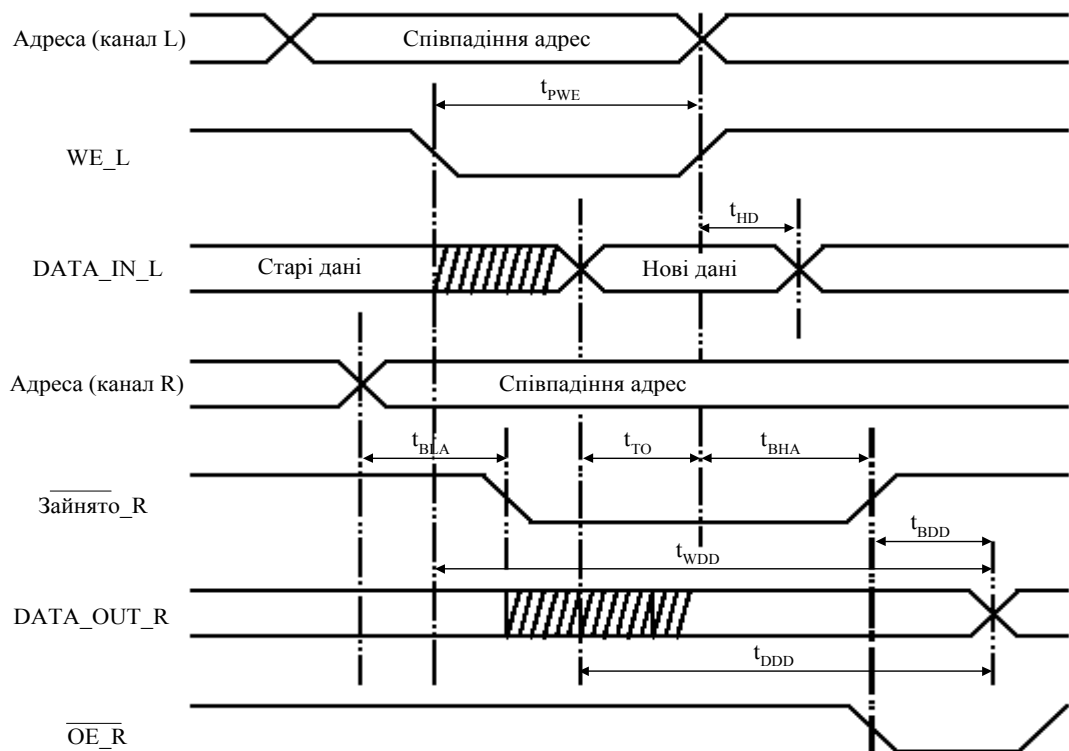


Рисунок 2.13 - Часові діаграми роботи схеми за методом зайнятості

Для методу зайнятості характерним є “вікно” тривалістю часу переключення порту  $t_{PS}=5ns$ , в межах якого існує невизначеність сигналу ЗАЙНЯТО, що приводить до пошкодження даних. Цю проблему можна рішити за допомогою генерації затримки циклу читання для порту, котрий програє арбітраж після аналізу переходу сигналу ЗАЙНЯТО з лог.1 в лог.0.

Недоліки методу зайнятості:

- додаткова операція обробки апаратного сигналу ЗАЙНЯТО в процесорах;
- додаткові цикли приводять до втрати швидкодії;
- ускладнення схеми для пристроїв пам'яті з кількістю портів більше ніж два;
- метод вимагає апаратної підтримки.

Схема доступу до пам'яті за методом переривань приведена на рисунку 2.14. Дві старші комірки пам'яті з адресою  $A_{p-1}$ ,  $A_p$  (двопортова пам'ять) зарезервовані для запису сигналів (прапорців) переривання (наприклад сигнал



ЗАЙНЯТО). Ця комірка пам'яті для процесора кожного порту виконує функцію поштової скриньки.

Операція запису з лівого порту в зарезервовану комірку пам'яті  $A_p$  генерує сигнал переривання  $INTR_R$  правого порту. Встановлення цього сигналу не забороняє виконання жодної операції. Операція читання правим портом комірки  $A_p$  скидає прапорець. Комірка з адресою  $A_{p-1}$  виконує такі ж самі функції поштової скриньки для лівого порту. Важливо відмітити, що операція читання скидає сигнал переривання, але самий вміст комірки пам'яті залишається без змін [14].

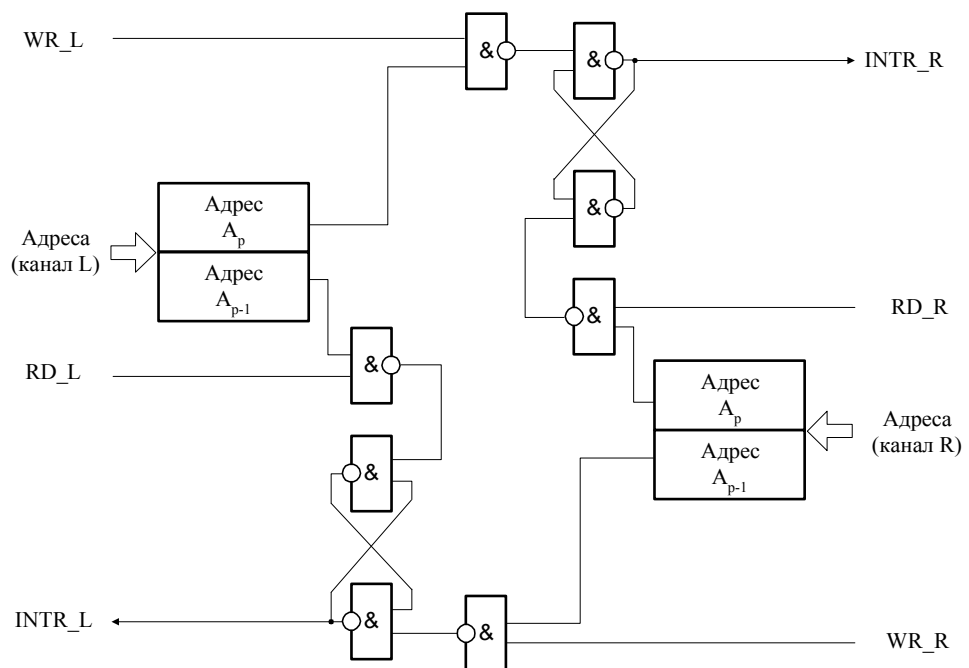


Рисунок 2.14 - Схема управління доступом до пам'яті за методом переривань

Якщо сигнал порту ЗАЙНЯТО є активним, то це забороняє порту пересилати сигнал переривання іншому порту. Окрім цього, сигнал ЗАЙНЯТО забороняє цьому порту читати значення власного поштової скриньки і таким чином скидати переривання. Якщо два порти звертаються за тією ж адресою одночасно, то пріоритетний порт здійснює арбітраж, в результаті якого один порт отримує контроль над цією коміркою. Варіанти доступу портів до пам'яті та результати арбітражу наведені в табл.2.1. Пояснення таблиці можна звести до розгляду трьох випадків:

- обидва порти читають дані;
- один порт читає, інший записує дані;
- обидва порти записують дані.

У першому випадку, незважаючи на читання однакових даних проводиться процедура арбітражу, в результаті якої визначається, який порт володіє усіма правами на цю комірку пам'яті. Порт, що програв контроль над цією коміркою, може читати дані, але сигнал ЗАЙНЯТО цього порту показує, що він програв арбітраж.

У другому випадку в результаті арбітражу буде встановлено пріоритет для порту, що записує дані. Прапорець ЗАЙНЯТО вказує, що “програшному” порту

Таблиця 2.1

Таблиця можливих варіантів доступу двох портів за методом переривань

№ п/п	Операція		Результат після арбітражу
	Лівий порт	Правий порт	
1	Читання	Читання	Обидва порти читають
2	Читання	Запис	Порту, що програв, забороняється записувати. Якщо ж “програвший” порт виконував читання і порти є асинхронні, то дані, що зчитуються можуть бути помилковими
3	Запис	Читання	- // -
4	Запис	Запис	-// -

заборонено запис даних і дозволено операцію читання. Для зчитаних даних в паралельній пам'яті існує поняття “старих”, “нових” або невизначеного статусу цих даних. В нашому випадку сигнал ЗАЙНЯТО вказує на “старі” дані.

В третьому випадку “програшному” порту забороняється операція запису, на що вказує прапорець ЗАЙНЯТО.

Таким чином, в методі переривань використовуються дві старші комірки пам'яті для генерації повідомлень іншому порту про характер процедури доступу до пам'яті (запис/читання). Цей метод арбітражу добре підходить для багатопортових буферних запам'ятовуючих пристроїв типу FIFO, але вимагає апаратної реалізації.

Управління доступом до пам'яті на основі методу семафорів може бути реалізоване програмним або апаратним шляхом [3]. У програмному методі пам'ять ділиться на блоки, для яких формуються правильний набір тестових операцій (семафорний доступ) перед наданням доступу одному з портів. Процесор може намагатись отримати контроль над блоком пам'яті за допомогою використання спеціальних інструкцій тестування стану семафора і встановлення семафора. Ці операції обов'язково мають виконуватись одна за одною, в іншому випадку можливе виникнення конфліктної ситуації. Якщо семафор є вільний, то процесор встановлює його і набуває контроль над блоком пам'яті, за який відповідає даний семафор.

Апаратна реалізація методу семафорів не вимагає від процесора набору спеціальних інструкцій для роботи з семафорами, використовується лише стандартна операція читання/запису в пам'ять, оскільки в даному методі доступу не вимагається блокування іншого процесору на час операцій читання/запису.

Технічно апаратна реалізація семафорної логіки базується на схемі символічної пересилки (рис.3.15). Ця схема дозволяє порту, що заволодів семафором, отримати доступ до розділеної області пам'яті. Позбутись володіння цією областю пам'яті порт може тільки за власної ініціативи. Запит порту на володіння семафором буде відкинутий, якщо семафором володіє вже інший порт.

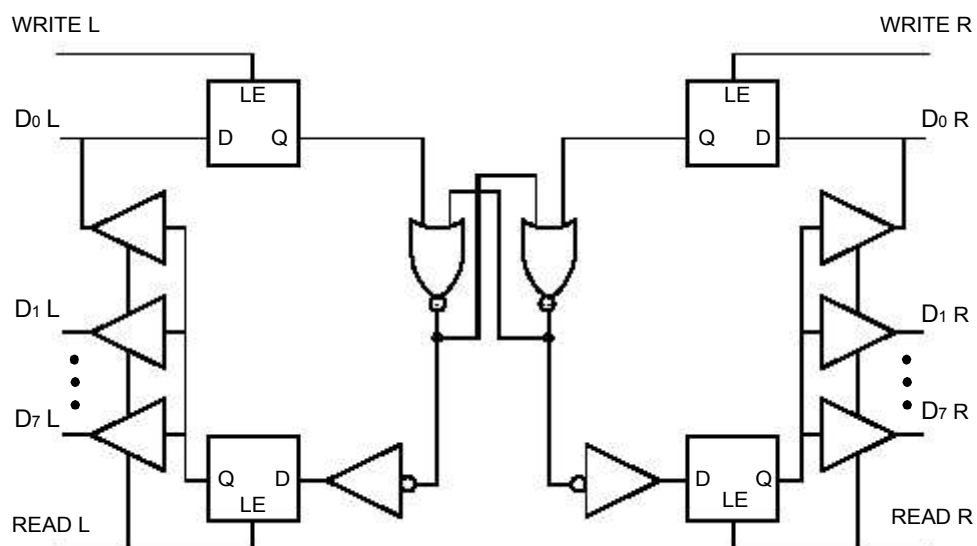


Рис. 3.15. Апаратна реалізація методу семафорів

Семафор будується на двох D-тригерах. Вхідні D-тригери застосовується для зберігання запитів щодо встановлення та очищення значень семафорів. Вихідні D-тригери використовуються, щоб заборонити можливість зміни семафору під час читання з порту. Семафорні комірки пам'яті адресуються подібно до звичайних комірок RAM. Можливі варіанти станів семафору зведено в табл.3.2.

Таблиця 2.2 – Таблиця станів семафорів

Функція	Лівий порт		Правий порт		Виконувана дія
	Запис	D <sub>0</sub>	Запис	D <sub>0</sub>	
Ніяких дій	1	1	1	1	Семафор вільний
L порт записує "0" в D <sub>0</sub>	0	0	1	1	L порт здобуває контроль
R порт записує "0" в D <sub>0</sub>	0	0	0	0	Без змін, L порт контролює блок пам'яті
L порт записує 1 D <sub>0</sub>	1	1	0	0	R порт здобув контроль
R порт записує 1 D <sub>0</sub>	1	1	1	1	Семафор вільний
L порт записує 0 D <sub>0</sub>	0	0	1	1	L порт здобуває контроль
L порт записує 1 D <sub>0</sub>	1	1	1	1	Семафор вільний

Семафорна логіка використовує довільну кількість спеціальних комірок пам'яті (семафорів) для управління заданою кількістю блоків багатопортової пам'яті. Цей вид логіки арбітражування добре підходить для багатопортових запам'ятовуючих пристроїв з блоковим методом запису/читання. Програмна реалізація семафорного арбітражування вимагає додаткових циклів звертання до області пам'яті семафорів.

### 3.7. Аналіз засобів моделювання

Моделювання та проектування даних і додатків ґрунтується на сформульованих вимогах і є вельми важливою частиною процесу створення готового продукту. Інструменти для підтримки даного етапу життєвого циклу програм можна умовно розділити на засоби моделювання, засоби проектування даних і засоби об'єктно-орієнтованого моделювання. Відзначимо, однак, що сьогодні багато компаній виробляють всі категорії інструментів, що інтегруються між собою (наприклад, дозволяють згенерувати модель даних і моделі процесів або синхронізувати їх між собою), або реалізують функціональність декількох різнотипних засобів моделювання в одному продукті, тому стосовно зазначеної категорії інструментів має сенс говорити про продукції різних виробників [2].

Нижче перераховані найбільш відомі продукти призначених для моделювання та проектування.

Мова VHDL служить для опису моделі цифрового пристрою (приладу, системи). Опис на мові VHDL визначає зовнішні зв'язки пристрою Мова опису апаратури для високошвидкісних інтегральних схем (VHSI), називаний VHDL, є формальним записом , що може бути використана на всіх етапах розробки електронних систем. Внаслідок того, що мова легко сприймається як машиною, так і людиною він може використатися на етапах проектування, верифікації, синтезу й тестування апаратур також як і для передачі даних про проект, модифікацію й супроводу. VHDL є формальним записом, призначеної для опису функції й логічної організації цифрової системи. Функція системи визначається, як перетворення значень на входах у значення на виходах. Причому час у цьому перетворенні задається явно. Організація системи задається переліком зв'язаних компонентів [4].

### 3.7.2. Аналіз AHDL

AHDL (мова опису апаратури фірми Altera) є високорівневою, модульною мовою, повністю інтегрованою в систему MAX+PLUS II. Він особливо добре підходить для проектування складної комбінаційної логіки, шин, кінцевих

автоматів, таблиць істинності і параметричної логіки. Ви можете використати текстовою редактор системи MAX+PLUS II або будь-який інший для створення текстових файлів проектів (AHDL Text Design Files (.tdf)). Потім Ви можете відкомпілювати TDF файли для отримання вихідних файлів, придатних для подальшого моделювання, тимчасового аналізу і програмування пристрою. Крім того компілятор системи MAX+PLUS II може створювати текстові файли експортування (AHDL Text Design Export Files (.tdx)) і текстові вихідні файли (Text Design Output Files (.tdo)), які можна зберегти як TDF файли і повторно використати в якості файлів проекту.

Оператори і елементи AHDL є потужним, багатогранним і легким у використанні засобом. Ви можете створювати увесь ієрархічний проект за допомогою AHDL або змішувати TDF файли з іншими типами файлів в один проект.

Хоча Ви можете скористатися будь-яким текстовим редактором для створення TDF файлів, але тільки текстовою редактор системи MAX+PLUS II дає Вам можливість скористатися його перевагами, коли Ви вводите, компілюєте і відлагоджуєте Ваш AHDL проект [6].

AHDL проект легко вставити в ієрархію проекту. У текстовому редакторі Ви можете автоматично створювати символ, представляючий TDF файл і вводити його в графічний файл проекту (Graphic Design File (.gdf)). Аналогічним чином Ви можете об'єднувати замовлені функції і понад 300 мегафункцій і макрофункцій, Альтерой, що поставляються, включаючи функції бібліотеки параметричних модулів (LPM), у будь-якій TDF файл.

Ви можете використати команди меню Assign або Assignment & Configuration File (.acf) для створення ресурсу і вибору пристрою. Ви можете також перевірити синтаксис і виконати повну компіляцію для відладки і прогону Вашого проекту. Будь-які помилки, що з'являються, автоматично локалізуються процесором повідомлень (Message Processor ) і виділяються у вікні текстового редактора.

### 3.7.3. Аналіз Verilog

Мова опису апаратних засобів Verilog (HDL) стала стандартом IEEE в 1995 як IEEE Std 1364-1995. Він був розроблений, щоб бути простим, інтуїтивним і ефективним на багатократних рівнях абстракції в стандартному текстовому форматі для різноманітності інструментів проекту, включаючи моделювання перевірки, розраховуючи аналіз, випробувальний аналіз, і синтез. Саме із-за цих багатих особливостей Verilog став вибором переважного числа проектувальників інтегральних схем (IC).

Verilog містить багатий набір вбудованих примітивів, включаючи логічні ворота, призначено-визначні примітиви, вимикачі, і пов'язану логіку. Він також має *pin - to - pin* затримки пристрою і перевірки, що розраховують. Змішування абстрактних рівнів по суті забезпечується семантикою двох типів даних : мережі і змінні. Тривалі привласнення, в яких вирази змінних і мереж можуть збільшуватися на мережі, забезпечують основну структурну конструкцію. Процедурні призначення, в яких результати обчислень, що залучають змінні і чисті цінності можуть бути збережені в змінні, забезпечують основну поведінкову конструкцію. Проект складається з ряду модулів, кожен з яких має вхід/вихід (I/O) інтерфейс, і опис його функції, яка може бути структурною, поведінковою, або з'єднання. Ці модулі сформовані в ієрархію і пов'язані з мережами [6].

Мова Verilog розширювана через інтерфейс мови програмування (PLI) і процедурний інтерфейс Verilog (VPI). PLI/VPI - набір рутинів, яке дозволяє іноземним функціям діставати доступ до інформації, що містилася в описі проекту Verilog HDL і полегшує динамічну взаємодію з моделюванням. Додатки PLI/VPI включають з'єднання Verilog HDL симулятора з іншою симуляцією і CAD -системами, налаштованими на завдання відладки, розрахунків, і коментатори. У 1990, Cadence Design Systems помістили Verilog HDL в громадську область, і незалежний, Open Verilog International (OVI) був сформований, щоб управляти і просувати Verilog HDL. У 1992, Рада директорів OVI почала спробу зробити Verilog HDL стандартом IEEE. У 1993, перша

робоча група IEEE була сформована; і після 18 місяців зосереджених зусиль, Verilog став стандартом IEEE як Станд. IEEE 1364-1995.

#### Висновки до розділу

1. Найшвидкодійною є паралельна пам'ять на основі багатовходових комірок пам'яті, а найменших апаратних затрат вимагає паралельна пам'ять з часовим розподілом ресурсів пам'яті.

2. Вибір базових структур для реалізації паралельної пам'яті з заданими технічними параметрами доцільно здійснювати за допомогою оцінки ефективності використання обладнання.

3. Аналітичні вирази для оцінки апаратних затрат, швидкодії та ефективності використання обладнання дозволяють оцінити основні характеристики компонентів і паралельної пам'яті з точністю достатньою для практичного використання.

4. Для погодження інтенсивності надходження даних з швидкістю пам'яті доцільно використовувати пристрої паралельно-послідовного та послідовно-паралельного перетворень.

5. Використання конвеєрних регістрів даних в контролері ПП дозволяє управляти доступом та зменшити час вибірки з пам'яті до часу спрацювання регістру.

6. Простота нарощування кількості портів і можливість адаптації ПП з часовим розподілом ресурсів до швидкодії зовнішніх пристроїв розширяє сфери її застосування.

7. Адаптація ПП до структур даних і алгоритмів розв'язання задач дозволяє зменшити апаратні затрати, підвищити швидкодію та ефективність використання обладнання.

8. Для забезпечення зміни з кроком одиниця величини ємності пам'яті FIFO необхідно її розділити за допомогою  $2^P$ -входових комутаторів на блоки регістрів величиною  $\sum_{g=0}^{P-1} 2^{(Pk-1)-g}$ . Ускладнення комутаторів веде до збільшення ємності блоків регістрів, кількості зв'язків і нерегулярності структури.



9. Серед основних методів управління доступом до паралельної пам'яті найшвидкодійним є метод зайнятості.

10. Вибір методів і базових структур для проектування паралельної пам'яті залежить від швидкодії запам'ятовуючого середовища та інтенсивності надходження даних.

### 3. РОЗРОБКА КОНЦЕПТУАЛЬНОЇ МОДЕЛІ БАГАТОПОРТОВОЇ ПАМ'ЯТІ

Для розробки багатопортової пам'яті необхідно побудувати її концептуальну модель. Концептуальна модель багатопортової пам'яті використовується для кращого розуміння поставленої задачі і пошуків шляхів її вирішення. Розроблена концептуальна модель наведена на рисунку 3.1.

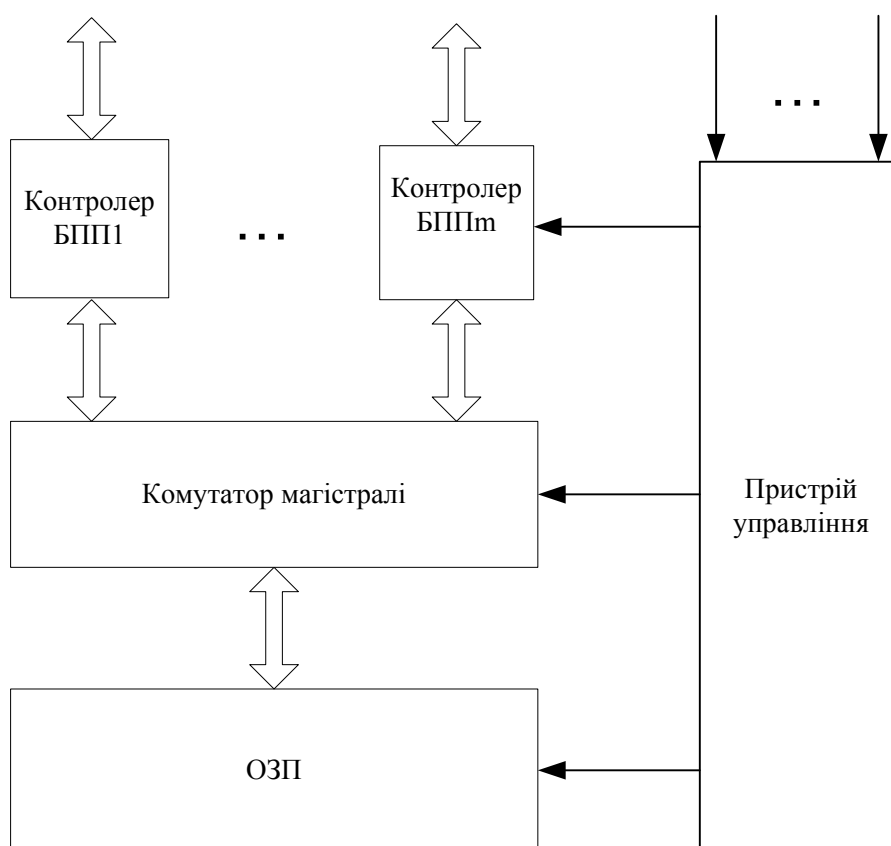


Рисунок 3.1 - Концептуальна модель багатопортової пам'яті.

Основними компонентами розробленої моделі є: контролери багатопортової пам'яті (БПП), пристрій управління (ПУ), комутатор магістралі (КМ) та одновимірне запам'ятовуюче середовище, яке реалізується на базі швидкодіючого оперативного запам'ятовуючого пристрою (ОЗП).

Така модель працює за алгоритмом фіксованих часових інтервалів, який передбачає для безконфліктного доступу виділення кожному пристрою фіксованого часового інтервалу у відповідності з наступною формулою:

$$T_{min} \geq m t_u \quad (3.1)$$

де  $T_{min}$  - найменший з періодів звертання пристроїв до БПП;  $t_u$  - цикл доступу до ОЗП;  $m$  – кількість пристроїв, що мають доступ до БПП. Виконання умови (1) дозволяє синтезувати пристрій обміну з інтенсивністю доступу, яка забезпечує роботу у реальному часі.

Основним елементом є ОЗП, вибір запам'ятовуючого середовища здійснюється залежно від структури даних та необхідної швидкодії обміну. ОЗП в основному визначає основні параметри пам'яті: швидкодію, ємність, ширину і варіанти паралельного доступу та має лінійну структуру, що забезпечує вирішення всіх проблем, що пов'язані з безконфліктністю одночасної паралельної вибірки елементів даних [7].

ПУ формує сигнали керування для ОЗП, налаштовує КМ на необхідні з'єднання між ОЗП та пристроями, що звертаються до пам'яті через формування неперервної послідовності тактових імпульсів для кожного контролера БПП. Однією з основних задач ПУ є синхронізація доступу до магістралі ОЗП зовнішніх пристроїв, які асинхронно працюють та вирішення конфліктів, що виникають в разі одночасного звернення двох або більше пристроїв до однієї комірки пам'яті.

Зв'язок між зовнішніми пристроями і ОЗП здійснюється через КМ та контролери БПП. Характеристики комутуючої мережі значною мірою впливають на характеристики БП. Для реалізації пристрою обміну було обрано комутатори магістралі, оскільки вони вимагають менших апаратних витрат, порівняно з комутаторами з просторовим розподілом або матричними комутаторами. Особливістю КМ є використання загальної інформаційної магістралі для передавання даних між ОЗП та контролерами БПП. Для розв'язання конфліктів при одночасному запиті на доступ до шини декількох пристроїв здійснюють призначення кожному пристрою унікального пріоритету

чи обслуговування за FIFO, або з'єднання пристроїв в ланцюжок чи коло з фіксацією їх пріоритетності. Для доступу до магістралі зазвичай використовується алгоритм фіксованих часових інтервалів.

Контролери БПП дозволяють скоротити час формування сигналів управління, генерації адреси та надходження даних в КМ. Для досягнення високої інтенсивності доступу до багатопортової пам'яті з часовим розподілом ресурсів одновимірного запам'ятовуючого середовища необхідно, щоб кількість контролерів БПП дорівнювала ширині доступу (кількості входів-виходів). Всі контролери БПП повинні бути однотипними та забезпечувати можливість роботи з різними за швидкістю ОЗП і зовнішніми пристроями; програмного налаштування на генерування необхідних послідовностей адреси; паралельного включення необхідної кількості контролерів БПП; виділення кожному контролеру БПП періодичних фіксованих часових інтервалів доступу до ОЗП [10]. Організація БП з використанням контролерів БПП дозволяє ефективно розподілити час доступу між різними за швидкістю зовнішніми пристроями.

### 3.1. Формування вимог і вибір принципів побудови багатопортової пам'яті

Об'єкти, процеси і явища реального світу, які опрацьовуються на комп'ютерних системах (КС) у своїй дискретній моделі практично завжди описуються певним чином організованою сукупністю даних. В більшості випадків вхідні дані мають просторово-часову організацію, яка визначається природою їх отримання [11]. Аналіз архітектур паралельних КС, структур даних і структур їх зберігання виявив наступні недоліки в архітектурно-структурній організації пам'яті:

- складність організації декількох звертань до внутрішньої пам'яті процесорів на протязі одного командного циклу;
- погана підтримка складних структур даних та відсутність багатовимірних запам'ятовуючих середовищ;
- програмна реалізація складних алгоритмів формування послідовностей адрес звертання до пам'яті;
- незбалансованість швидкодії пам'яті з швидкодією операційних пристроїв і процесорів;
- відсутність множинного доступу до зовнішньої пам'яті програм і даних на протязі командного циклу.
- При проектуванні пам'яті сучасних паралельних КС необхідно усунути вказані недоліки та забезпечити відповідність пам'яті наступним **вимогам**:
- мати організацію, що орієнтована на роботу з векторами і матрицями даних;
- забезпечити одночасне введення вхідного масиву даних і виведення раніше прийнятого масиву, дані в якому розміщені у відповідності до алгоритму розв'язання задачі;
- забезпечувати введення і виведення даних як з однаковою, так і з різною тактовою частотою;

- виконувати функції переставляння та затримки даних на необхідне число тактів;
- забезпечувати багатоканальне введення та виведення даних, причому число входів і виходів може бути різним;
- бути адаптованою до структури даних і специфіки задач, які розв'язуються.

Принципи побудови багатопортової пам'яті (БПП). Забезпечити вищевказані вимоги та покращити характеристики пам'яті можна шляхом використання наступних принципів:

- ієрархічності і багаторівневості побудови пам'яті великої ємності з широким використанням кешування і механізмів управління сигналами очікування;
- просторового і функціонального розділення внутрішньої пам'яті з широким використанням різних за швидкодією і шириною доступу модулів пам'яті;
- багатоканального доступу до пам'яті з метою забезпечення декількох звертань на протязі одного командного циклу;
- апаратної реалізації складних алгоритмів генерації послідовності адрес з використанням модульної арифметики;
- гармонічного поєднання можливостей інтегральної технології з розширенням функцій пам'яті, збільшення кількості внутрішніх і зовнішніх каналів доступу та з покращенням параметрів модулів пам'яті (ємності, швидкодії і т.д.).

### 3.2. Розробка структури багатопортової пам'яті

Поява на ринку НВІС пам'яті з великим об'ємом та малим часом циклу запису і читання, спонукала до розробки на їх базі БПП, в основу роботи якої покладено принцип часового розподілу ресурсів пам'яті між пристроями, що під'єднуються до неї. Для КС реального часу найбільший інтерес становлять безконфліктні методи доступу до БПП. Структура БПП, яка реалізує дані методи обміну, приведена на рисунку 3.1, де КБПП – контролер БПП, ПК – пристрій керування. Основними елементами БПП із часовим розподілом ресурсів є ОЗП і контролери БПП, які синхронізують обміном між ОЗП і зовнішніми пристроями.

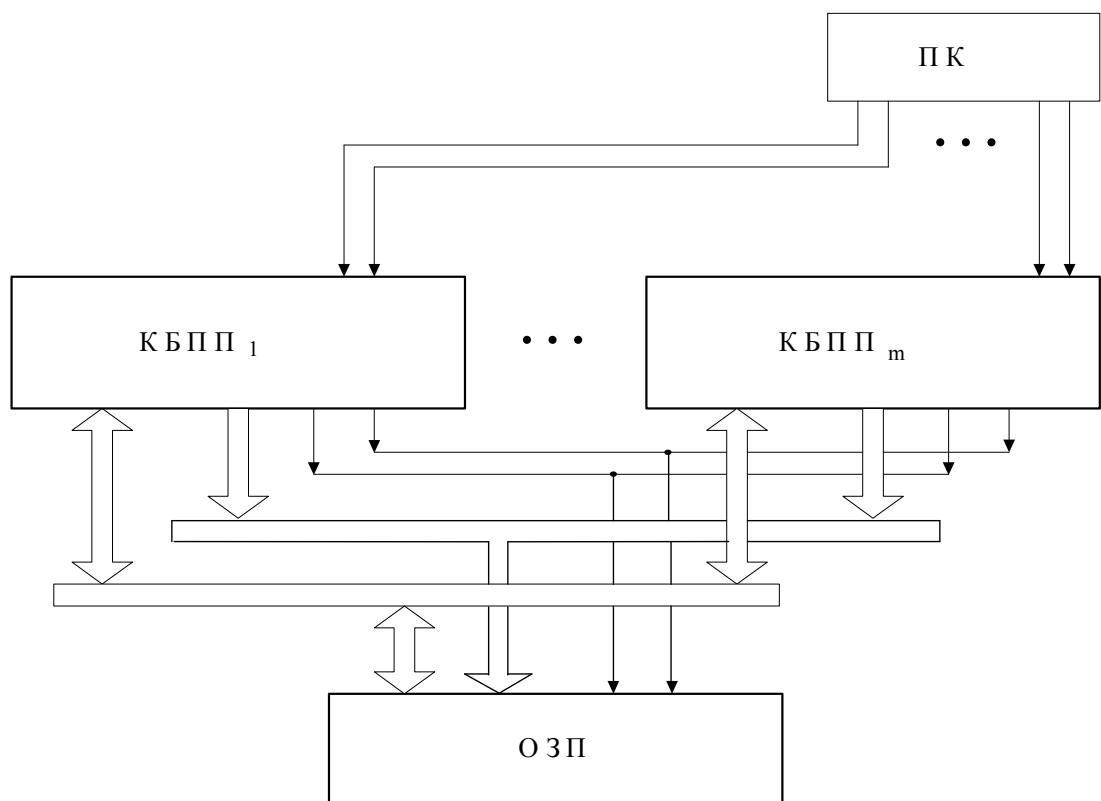


Рисунок 3.1. Структура БПП з часовим розподілом ресурсів

Кількість контролерів в БПП дорівнює кількості зовнішніх пристроїв. Доступ до ОЗП зовнішніх пристроїв здійснюється за алгоритмом фіксованих

часових інтервалів, за яким кожному КБПП циклічно з певним періодом  $T$  надає фіксований час доступу до ОЗП [9].

Період  $T$  звертання пристроїв до ОЗП залежить від їх кількості  $m$  і від часу  $t_{\text{ц}}$  циклу читання (запису) в ОЗП. Для зовнішніх пристроїв з однаковою швидкістю виконання операцій вводу-виводу період  $T$  визначається виразом:

$$T = t_{\text{ц}} m.$$

Налаштування БПП на роботу з різними за швидкістю ОЗП і пристроями здійснюється шляхом формування ПК для кожного КПП <sub>$j$</sub>  неперервної послідовності тактових імпульсів  $TI_{1j}, TI_{2j}$ , де  $j=1, \dots, m$ .

### 3.3. Розробка компонентів багатопортової пам'яті

Пристрій керування БПП. Структура пристрою керування ПК, який формує послідовності імпульсів для КБПП, наведена на рис.5.2, де  $TI$  – тактові імпульси,  $Tг$  – тригер,  $РгУ$  – регістр управління;  $Км$  – комутатор,  $ПУ$  – вхід початкового устанавлення,  $УпрКм$  – вхід правління комутатором. Однією із основних задач вузла ПК є синхронізація доступу до магістралі ОЗП асинхронно працюючих зовнішніх вузлів КС та формування необхідної неперервної послідовності тактових імпульсів  $TI_{1j}, TI_{2j}$ .





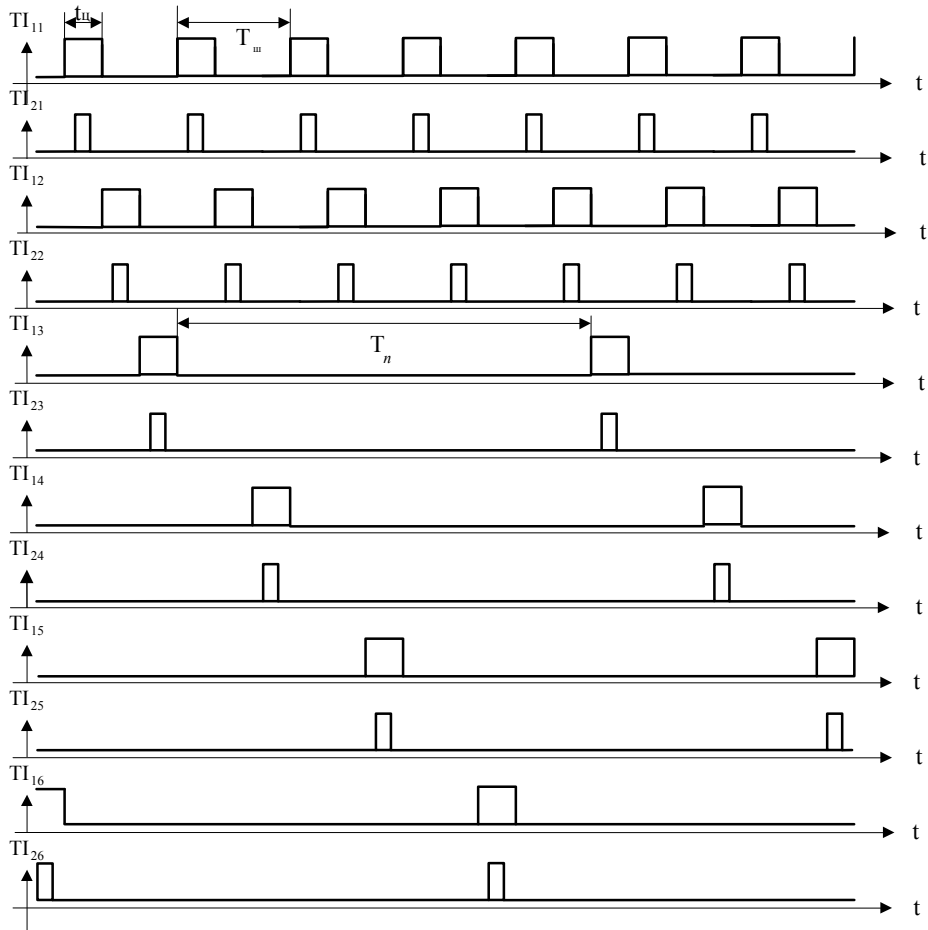


Рисунок 3.3 - Часова діаграма формування тактових імпульсів для КПП

Тривалість тактових імпульсів  $\Pi_{1j}$ ,  $\Pi_{2j}$  рівна, відповідно, часу циклу звертання до ОЗП і часу запису (читання) в ОЗП. Коли до ПП підключаються пристрої різні за швидкодією, то періоди звертання  $T$  для  $L$ -швидкодійних пристроїв будуть рівні:

$$T_{ш} = (L+1)t_{ц},$$

а для повільних:

$$T_{п} = T_{ш}(n-L).$$

На рисунку 3.3 приведена часова діаграма формування тактових імпульсів  $\Pi_{1j}$ ,  $\Pi_{2j}$  для управління контролерами паралельної пам'яті з загальною кількістю пристроїв  $m=6$ , з яких два швидкі  $L=2$ .

Контролер ПП генерує необхідну послідовність адрес і формує сигнали управління для доступу до ОЗП. Основні параметри БПП у значній мірі

залежать від параметрів контролера БПП. Основним шляхом, що дозволяє значно зменшити габарити, потужність споживання та підвищити надійність БПП є реалізація КПП у вигляді спеціалізованої НВІС. Для цього структура контролера повинна бути орієнтована на реалізацію у вигляді НВІС та забезпечувати:

- налаштування за допомогою зовнішніх тактових імпульсів на роботу з різними за швидкістю ОЗП і зовнішніми пристроями;
- програмне налаштування на генерацію необхідної послідовності адрес;
- мінімальний час вибірки даних з ПП;
- нарощування кількості пристроїв, що підключаються до ОЗП.

Структура КБПП, що відповідає таким вимогам, приведена на рисунку 3.4. Основними елементами даної структури є буферні регістри Pr1 і Pr2, генератор адреси ГА, схеми формування сигналів управління Ф1, Ф2 і Ф3 та шинні формувачі.

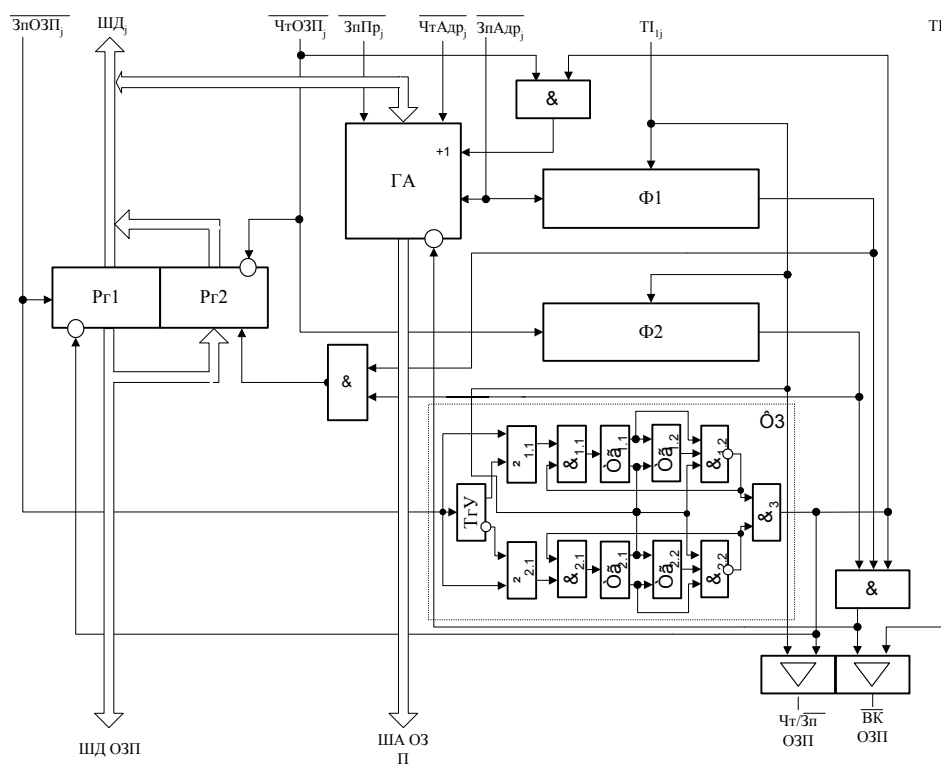


Рисунок 3.4 - Структурна схема контролера БПП

Буферні регістри Pr1 і Pr2 дозволяють організувати двонаправлену передачу даних, синхронізувати доступ до ОЗП та провести часовий розподіл

його ресурсів. Схеми формування сигналів управління  $\Phi_1, \Phi_2, \Phi_3$  призначені для фіксації сигналів управління  $ЗпОЗП_j, ЧтОЗП_j, ЗпАдр_j$ , які поступають від зовнішнього  $j$ -го пристрою та запуску формування в визначений час одиничних імпульсів від'ємної полярності тривалістю приблизно рівною тривалості імпульсів, що поступають із тактового входу  $ТI_{1j}$ . Шинні формувачі  $КПП_j$  у момент надходження тактового сигналу  $ТI_{1j}$  видають сигнали управління  $Чт/ \bar{Зп} ОЗП, \bar{ВК} ОЗП$  на входи  $ОЗП$ . У цей же момент часу на вхід дозволу  $ГА$  надходить сигнал, який забезпечує видачу адреси з  $ГА$  на входи шини адреси  $ОЗП$  [5].

Для реалізації запропонованого контролера БПП необхідні наступні витрати обладнання:

$$W_{КТР} = 2W_{Pr} + W_{ГА} + 3W_{\Phi} + 2(W_I + W_{ШФ}),$$

де  $W_{Pr}, W_{ГА}, W_{\Phi}, W_I, W_{ШФ}$  - апаратурні витрати на реалізацію відповідно регістра, генератора адрес, схеми формування сигналів управління, логічного елемента  $I$  та шинного формувача. Оскільки контролер ПП зорієнтований на реалізацію у вигляді НВІС, то для оцінки витрат обладнання на його реалізацію за одиницю виміру доцільно взяти логічний вентиль. Апаратурні витрати на реалізацію розробленого контролера ПП у вигляді спеціалізованої НВІС у вентилях визначається виразом:

$$W_{КТР} = 18g + 69p + 2p^2 + 44,$$

де  $g$  - розрядність шини даних,  $p$  - розрядність шини адреси.

Схема  $ГА$  наведена на рисунку 3.5, де Ліч - лічильник; КС - програмоване універсальне комутуюче середовище; БКІ - блок керованих інверторів; ПУ - пристрій управління; С<sub>м</sub> - суматор.

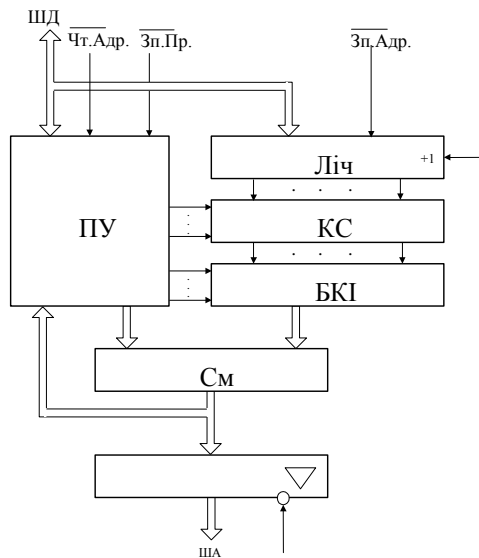


Рисунок 3.5 - Структурна схема генератора адрес

Такий набір елементів забезпечує підрахунок даних, що записуються або читаються, перестановку й інвертування розрядів лічильника та додавання їх до адреси зміщення.

Налаштування ГА на генерацію необхідного порядку адрес здійснюється програмним шляхом. Максимальна кількість тактів програмування визначається розрядністю адреси. В кожному такті програмування здійснюється комутація одного із виходів блоку КС з виходом відповідного розряду лічильника та задається спосіб його передачі (у прямому або інверсному коді) на суматор [10].

### 3.4 Синтез багатопортової пам'яті для комп'ютерних систем реального часу

Основною метою синтезу БПП для КС реального часу є отримання орієнтованої на НВІС-реалізації структури, яка має високу ефективність використання обладнання та забезпечує доступ до пам'яті з інтенсивністю надходження даних. Вихідною інформацією для синтезу БПП комп'ютерних систем реального часу є:

- узгоджений потоковий граф алгоритму розв'язання задачі;

- кількість каналів надходження даних;
- величина масивів даних;
- розрядність даних;
- максимальна частота звертання кожного каналу до пам'яті;
- базові структури паралельної пам'яті;
- техніко-економічні вимоги та обмеження.

На основі аналізу узгодженого потокового графу алгоритму розв'язання задачі та вихідної інформації визначаються алгоритми формування послідовностей адреси, ємність та інтенсивність доступу до паралельної пам'яті. Для синтезу БПП з високою ефективності використання обладнання необхідно узгодити інтенсивність надходження даних з інтенсивністю доступу до паралельної пам'яті. Інтенсивність доступу до паралельної визначається так:

$$P_{\text{ПП}} = m_{\text{ПП}} n_{\text{к}} F_{\text{ПП}}, \quad (3.1)$$

де  $t_{\text{ПП}}$  – ширина доступу до паралельної пам'яті,  $n_{\text{к}}$  – розрядність каналу доступу до паралельної пам'яті,  $F_{\text{ПП}}$  – частота доступу до паралельної пам'яті. З формули (5.1) видно, що інтенсивність доступу до паралельної пам'яті залежить як від ширина і частоти доступу, так і розрядності каналів. Вибір і оцінку синтезованих структур БПП пропонується здійснюється за інтегральним параметром ефективністю використання обладнання, який визначається так:

$$E = \frac{m_{\text{ПП}} n_{\text{к}} Q}{W_{\text{ПП}} \times t_{\text{цПП}}},$$

де  $W_{\text{ПП}}$  - затрати обладнання на реалізацію паралельної пам'яті;  $t_{\text{ПП}}$  – ширина доступу до паралельної пам'яті;  $Q$  – ємність паралельної пам'яті;  $t_{\text{цПП}}$  – цикл доступу до паралельної пам'яті. Такий інтегральний параметр зв'язує між собою ємність, інтенсивність доступу і витрати обладнання на реалізацію паралельної пам'яті та дає оцінку вкладу кожного елемента (вентиля) схеми у зберігання інформації та забезпечення інтенсивності доступу пам'яті.

При синтезі БПП необхідно враховувати дві діалектичні протилежності – універсалізацію та спеціалізацію. Універсалізація розглядається як засіб скорочення номенклатури, зменшення затрат на проектування та виготовлення,

а спеціалізація – як засіб досягнення високої ефективності використання обладнання за рахунок адаптації структури паралельної пам'яті до алгоритмів обробки та структури даних. Розглянемо методи синтезу паралельної пам'яті на основі базових структур з елементами універсалізації [7].

Інтенсивність надходження даних, час періоду (циклу) доступу до запам'ятовуючого середовища та алгоритми формування послідовностей адреси при доступі до пам'яті визначають методи проектування паралельної пам'яті. Для проектування паралельної пам'яті ОС реального часу використаємо метод просторово-часового у відображення структури потоків даних у узгодженому потоковому графі алгоритму розв'язання задачі на базовій структурі паралельної пам'яті. На основі вимог конкретних застосувань вибирається структура БПП та засоби узгодження інтенсивності надходження даних і доступу до пам'яті.

Інтенсивність доступу до БПП, яка визначилася на основі аналізу узгодженого потокового графу алгоритму розв'язання задачі, є визначальною при виборі базової структури паралельної пам'яті та методу доступу до запам'ятовуючого середовища. БПП і метод часового розподілу ресурсів запам'ятовуючого середовища використовується для синтезу БПП у випадку виконання наступної умови:

$$F_{\text{ПП}} m_{\text{ПП}} n_{\text{ПП}} \leq n_{\text{ЗС}} F_{\text{ЗС}}, \quad (3.2)$$

де  $F_{\text{ПП}}$  - частота звертання пристроїв до паралельної пам'яті;  $F_{\text{ЗС}}$  - частота доступу до запам'ятовуючого середовища;  $m_{\text{ПП}}$  – ширина доступу до паралельної пам'яті,  $n_{\text{ПП}}$  – розрядність каналу доступу до паралельної пам'яті;  $n_{\text{ЗС}}$  – розрядність запам'ятовуючого середовища. Виконання умови (3.2) дозволяє синтезувати паралельну пам'ять з інтенсивністю доступу, яка забезпечує роботу у реальному часі.

Метод доступу на основі послідовно-паралельного перетворення та розшарування запам'ятовуючого середовища використовується для наступного випадку:

$$F_{\text{ПП}} m_{\text{ПП}} n_{\text{ПП}} \geq n_{\text{ЗС}} F_{\text{ЗС}}. \quad (3.3)$$

Виконання умови (5.3) вимагає зміни параметрів запам'ятовуючого середовища для забезпечення необхідної інтенсивності доступу до паралельної пам'яті. Одним із шляхів збільшення інтенсивності доступу до паралельної пам'яті є використання послідовно-паралельного перетворення та збільшення розрядності запам'ятовуючого середовища шляхом його розшарування  $K$  незалежних сегментів (модулів пам'яті). Величина  $K$  визначається відношенням:

$$K = \left\lceil \frac{F_{\text{ПП}} n_{\text{ПП}} m_{\text{ПП}}}{F_{\text{ЗС}} n_{\text{ЗС}}} \right\rceil,$$

де  $\lceil$ -знак округлення до більшого цілого. При розшаруванні запам'ятовуючого середовища на  $K$  незалежних сегментів, це дозволяє збільшити розрядність запам'ятовуючого середовища у  $K$  раз.

Для випадку, коли:

$$F_{\text{ПП}} m_{\text{ПП}} n_{\text{ПП}} > n_{\text{ЗС}} F_{\text{ЗС}} > F_{\text{ПП}} n_{\text{ПП}}$$

для побудови паралельної пам'яті використовується метод часового розподілу ресурсів та доступ на основі послідовно-паралельного перетворення запам'ятовуючого середовища. При цьому для забезпечення необхідної інтенсивності доступу до паралельної пам'яті необхідно розшарувати його запам'ятовуючого середовища. Кількість сегментів, які необхідні для забезпечення необхідної інтенсивності доступу, визначається за наступною формулою:

$$K \geq \left\lceil \frac{t_{\text{чЗС}} \times m_{\text{ПП}}}{T_{\text{min}}} \right\rceil$$

Використання комбінації методу часового розподілу ресурсів та доступу на основі послідовно-паралельного перетворення запам'ятовуючого середовища дозволяє синтезувати БПП з високою ефективністю використання обладнання.



### 3.6. Моделювання компонентів БПП

Моделювання контролера БПП здійснювалось на мові VHDL в середовищі Active-HDL 9.1. Пакет Active-HDL - це повністю інтегроване середовище розробки цифрових пристроїв на базі текстових описів і графічних схем. Active-HDL є провідним середовищем для створення проектів та їх моделювання для широкого спектру програмованих логічних інтегральних схем (ПЛІС / FPGA), забезпечуючи гнучкість підходу і надаючи розвинені функції підтримки найбільш складних сучасних проектів. Середовище проектування Active-HDL дає розробникам незалежність у використанні інших засобів проектування з єдиного, повністю інтегрованого оточення.

На рисунку 3.6. представлена схема контролера БПП. Елементи U1, U2, U3 – відповідають схемам формування сигналів управління Ф1, Ф2, Ф3; U4 – генератор адреси ГА; U5 та U6 відповідають буферним регістрам Rг1 та Rг2; BF1 і BF2 – шинні формувачі. Також на схемі є три логічні елементи «І».

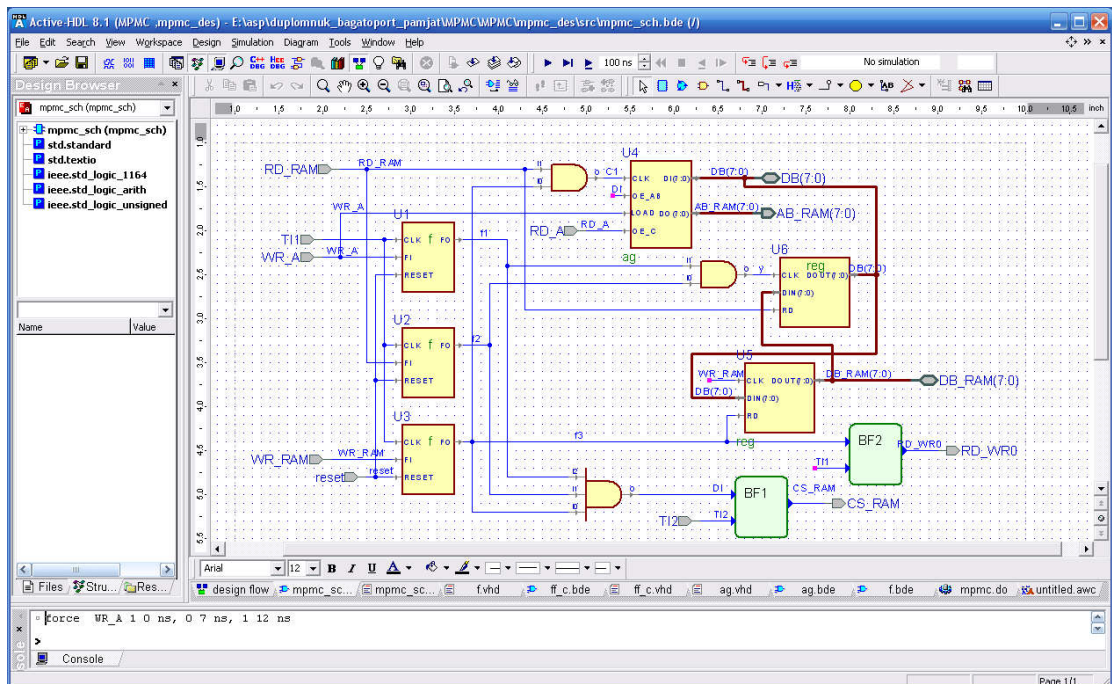


Рисунок 3.6 - Схема контролера БПП

Вхідними сигналами є: WR\_RAM – це сигнал запису даних ЗпОЗП, DB – шина даних, RD\_RAM – сигнал читання даних ЧтОЗП, RD\_A – сигнал читання адреси ЧтАдр, WR\_A – сигнал запису адреси ЗпАдр, T11– імпульси, що поступають з тактового входу, T12 – сигнал входу вибірки. Вихідні сигнали: DB\_RAM – відповідає сигналу шини даних ШД ОЗП, AB\_RAM – відповідає шині адрес ША ОЗП, RD\_WR0 – відповідає сигналам читання/запису Чт/Зп ОЗП, CS\_RAM – вибір кристалу ВК ОЗП. Шини DB, DB\_RAM та AB\_RAM вибрано восьми розрядними [6].

На рисунку 3.7. представлена часова діаграма контролера БПП.

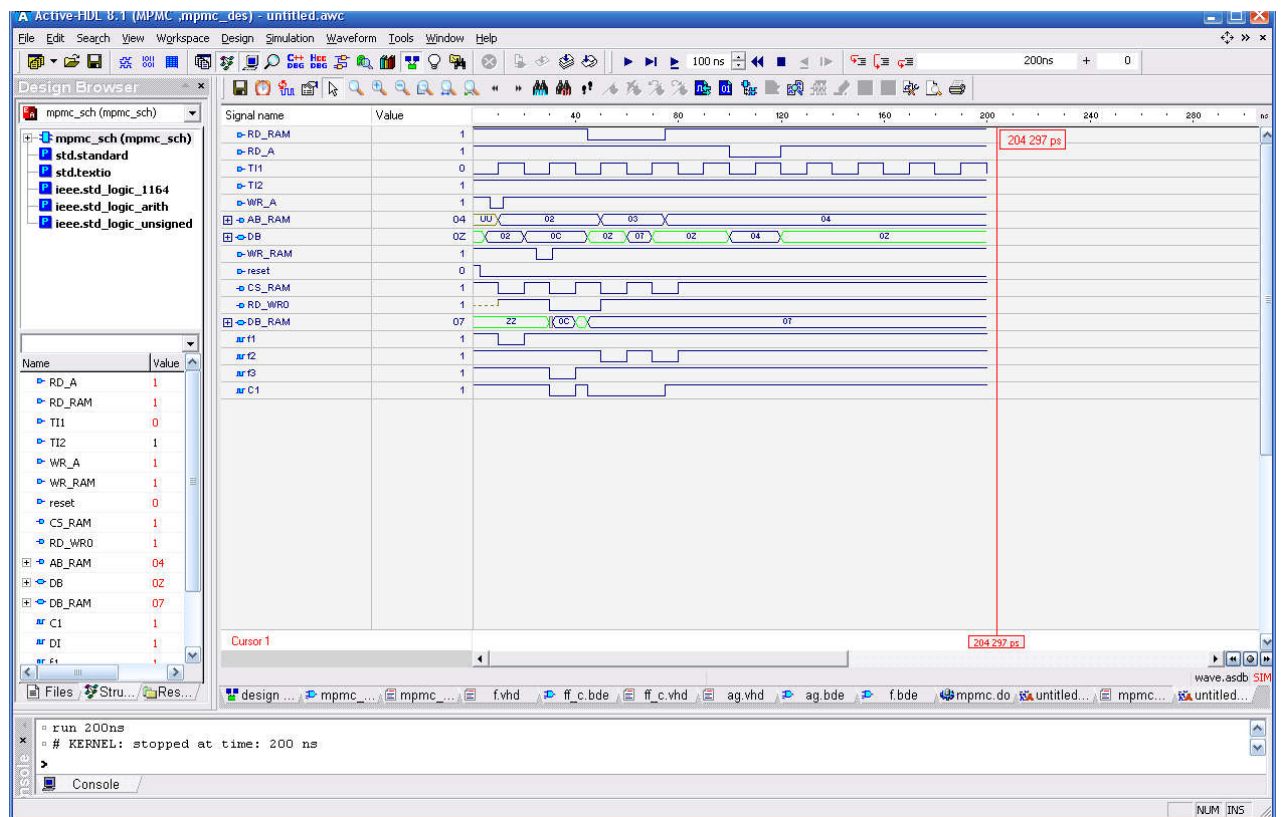


Рисунок 3.7. Часова діаграма контролера БПП

Контролер БПП може виконувати такі операції: попередній запис адреси, читання інформації з паралельної пам'яті, запис інформації в паралельну пам'ять за раніше записаною адресою та читання вмісту адреси зовнішнім пристроєм.

Попередній запис адреси здійснюється за імпульсом від'ємної полярності WR\_A з DB (на який адреса поступає зовнішнім пристроєм і в даному випадку

становить 02) в генератор адрес - U4. Імпульс фіксується схемою формування Ф1. Сигнал f1, який формується на виході схеми формування Ф1, по передньому фронту передає значення що містились в U4 на AB\_RAM. Отже в результаті виконання операції, адреса (що становить 02 в даному випадку) поступає на AB\_RAM.

Читання інформації з пам'яті здійснюється за імпульсом від'ємної полярності входу RD\_RAM. Переднім фронтом цього сигналу здійснюється збільшення лічильника U4 на 1. При проходженні тактового імпульсу T11, формується сигнал f2 на схемі формування Ф2, який підключає на AB\_RAM виходи генератора адрес U4 та змінює значення сигналу RD\_WR0 з 0 на 1, тобто на дозвіл читання. Інформація зчитується з DB\_RAM (в даному випадку становить - 07) та записується в U6, а звідти зчитується в DB за переднім фронтом сигналу f2.

Запис інформації в паралельну пам'ять за раніше записаною адресою здійснюється переднім фронтом сигналу WR\_RAM. Дані, які необхідно записати, знаходяться в DB (в даному випадку це – 0C) і за цим сигналом переходять в U5. Після приходу тактового імпульсу T11 на схему формування Ф3, на її виході утворюється сигнал f3. Він підключає на AB\_RAM вихід з лічильника U4 (попередньо збільшивши його вміст на 1) та U5 на DB\_RAM (на який переходять дані з DB). І, отже, на сигналі DB\_RAM в даному випадку встановлюється значення 0C, за сигналом RD\_WR0, який встановлюється в 0 (дозвіл запису) переднім фронтом сигналу f3 [12].

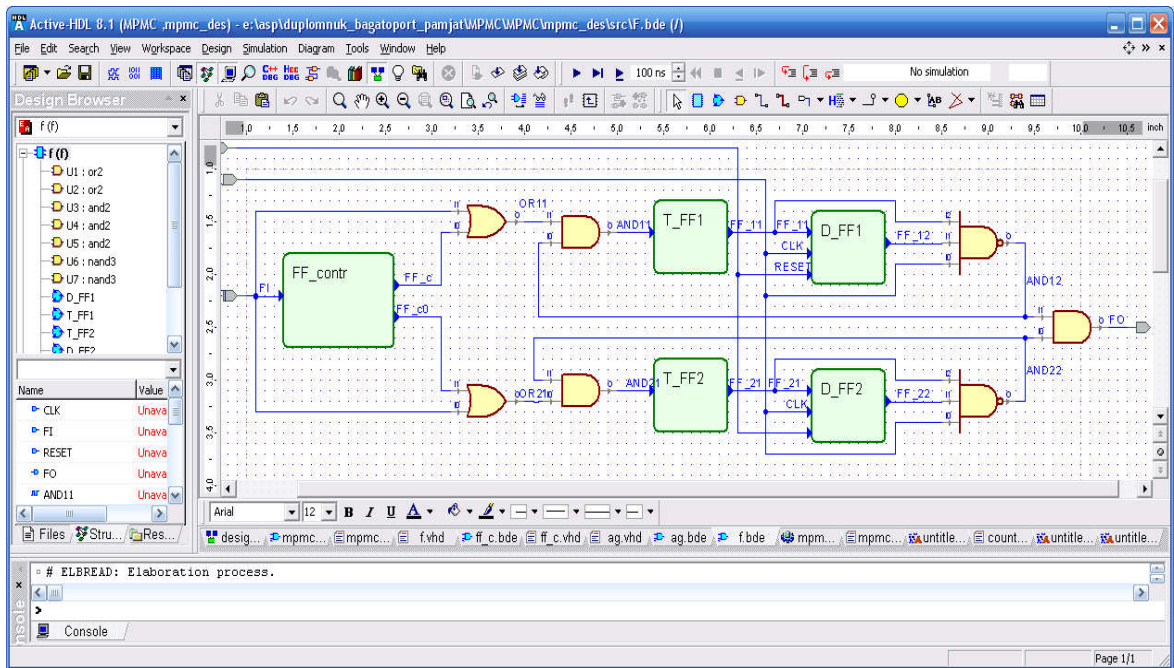


Рисунок 3.8 - Схема формувачів сигналів управління

Читання вмісту адреси зовнішнім пристроєм здійснюється за від'ємним імпульсом сигналу RD\_A, що дозволяє передачу адреси з лічильника AG (якій відповідає значення на AB\_RAM) на DB (в данному випадку значення - 04) через шинний формувач AG.

На рис.5.8. показана схема формувачів сигналів управління. Вона складається з: тригера управління - FF\_contr, 2 елементів «АБО», 3 елементів «І», 2 Т-тригерів - T\_FF1 та T\_FF2, 2 D-тригерів - D\_FF1 та D\_FF2 та 2 елементів «І-НЕ».

Часова діаграма формувачів сигналів управління наведена на рисунку 3.9. Основними сигналами є: F1 – вхідний сигнал, FF\_c, FF\_c0 – вихідні сигнали тригера управління (FF\_contr), OR11, OR21 – вихідні сигнали елементів «АБО», AND11, AND21 - вихідні сигнали елементів «І», FF\_11, FF\_21 - вихідні сигнали Т-тригерів - T\_FF1 та T\_FF2 відповідно, FF\_12, FF\_22 - вихідні сигнали D-тригерів - D\_FF1 та D\_FF2 відповідно, AND12, AND22 - вихідні сигнали елементів «І-НЕ», CLK – сигнал що надходить з тактового входу TI1, F0 – вихідний сигнал.

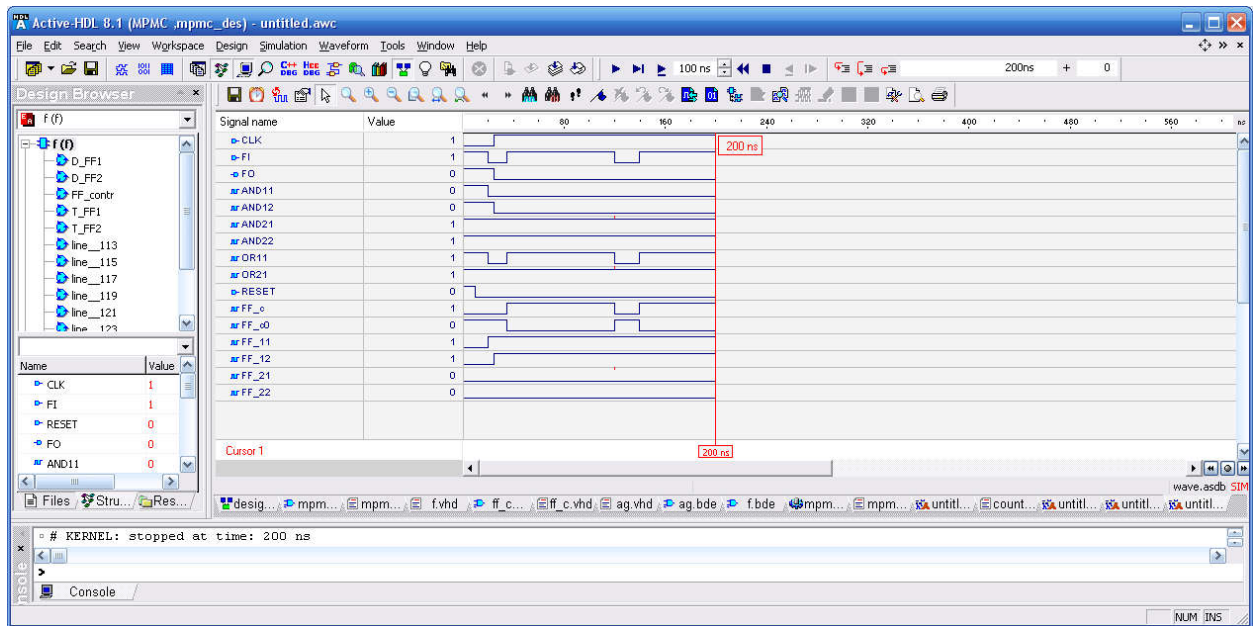


Рисунок 3.9 - Часова діаграма формувачів сигналів управління

На рисунку 3.10 показана схема генератора адрес. Основними елементами якої є: U1 – лічильник, перший шинний формувач – BF\_1 та другий шинний формувач BF.

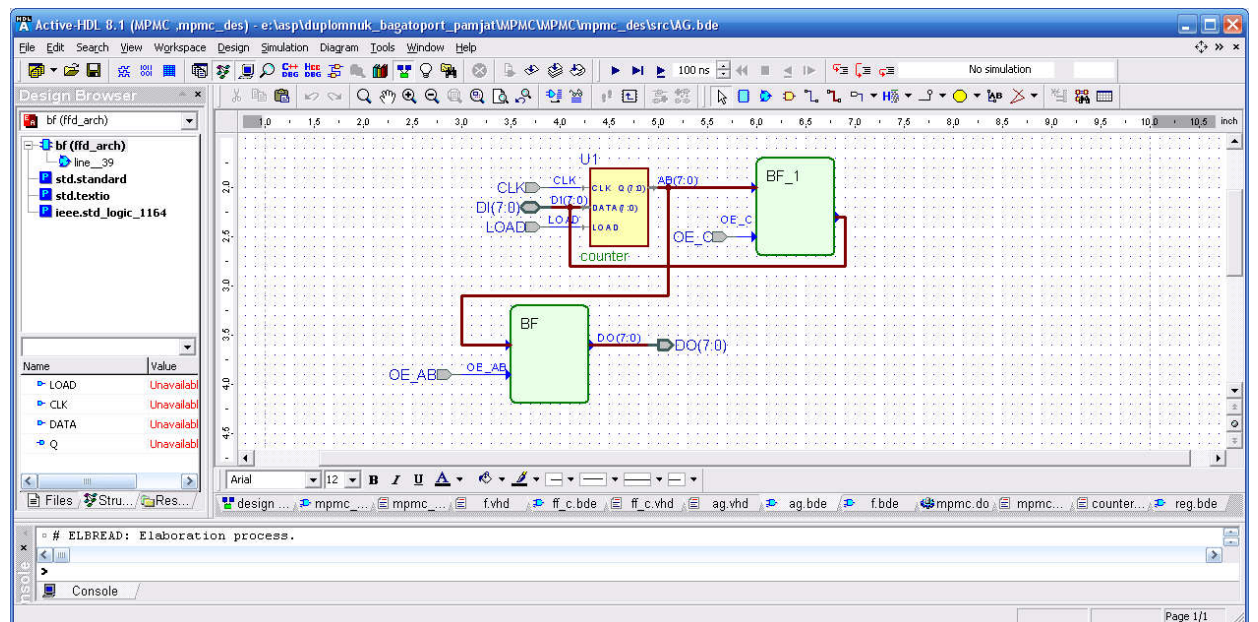


Рисунок 3.10. Схема генератора адрес.

Часова діаграма генератора адрес наведена на рисунку 3.11. Основними сигналами є: CLK – синхровхід лічильника адреси, що зв'язаний з виходом елемента «I» на схемі контролера БПП, DI – сигнал, що йде з шини даних (DB), LOAD – сигнал, що йде з WR\_A, AB – вихід лічильника, який пов'язаний з

входами шинних формувачів BF\_1 та BF, OE\_C – вхідний сигнал першого шинного формувача BF\_1, що йде з RD\_A, OE\_AB – вхідний сигнал другого шинного формувача BF, що йде з елемента «I», в який входять виходи формувачів сигналів управління, D0 – вихідний сигнал, що відповідає сигналу AB\_RAM на схемі контролера БПП [8].

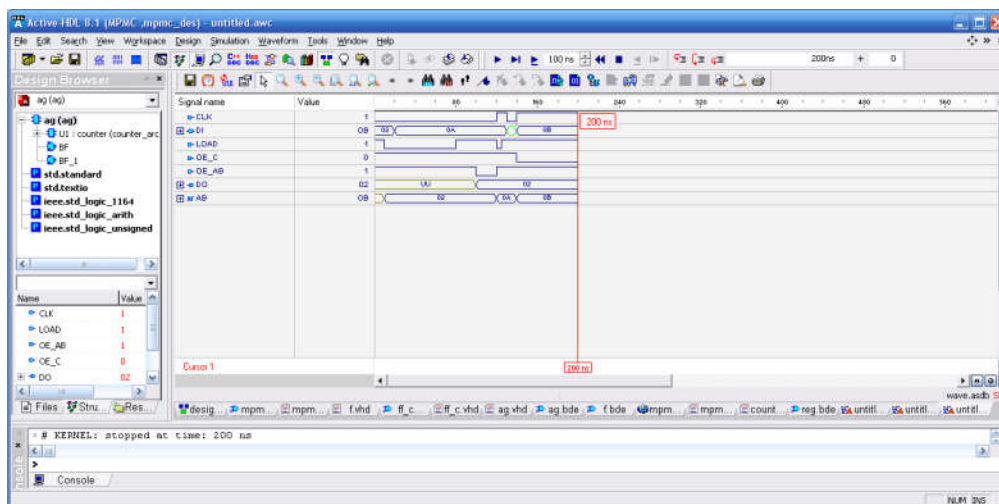


Рисунок 3.11. Часова діаграма генератора адрес

### 3.7. Розробка програми прикладу використання БПП в паралельно-потоківих системах

Структура паралельно-потоківих систем (ППС), показана на рисунку 3.12, містить  $K \times m$  процесорів цифрової обробки сигналів (ПЦОС), які розділені БПП на  $K$  сходинок [3]. Для ефективного розв'язання задач в такій системі, задачу необхідно розбити на  $K$  підзадач. При цьому кожна  $i$ -та підзадача ( $i=1, \dots, K$ ) паралельно вирішується  $m$  ПЦОС  $i$ -ої сходинки системи. Час вирішення  $i$ -ої підзадачі повинен бути рівний або менший такту роботи системи, який визначається виразом  $T \leq N / (F \times g)$ . При цьому, об'єм обчислень для кожної підзадачі не повинен перевищувати:

$$Q_i = \sum_{j=1}^m P_j \times T$$

де  $P_j$ -продуктивність  $j$ -го процесора.

Для обробки потоків даних в реальному масштабі часу за алгоритмами, що мають складність  $R$  продуктивність  $P$  системи ЦОС повинна бути

$$P=gFR/N,$$

де  $g$  - кількість вхідних інформаційних каналів;  $F$  - частота поступлення даних у вхідних інформаційних каналах,  $N$  – розмірність вхідного масиву.

Перед початком роботи ППС необхідно здійснити її налаштування на розв'язання певного кола задач. Ця процедура виконується шляхом завантаження в пам'ять програм процесорних модулів відповідних підпрограм, які є складовими частинами загальної програми. Процес обробки інформації починається з завантаження першого масиву вхідних даних і інформації управління, що супроводжує цей масив, в перший модуль БПП. В інформації управління вказується код задачі, яку необхідно виконати, розмірність і стан масиву. Інформація управління записується в визначенні адреси модуля БПП. Об'єм пам'яті модулів БПП визначається розмірами масивів даних, які обробляються, та інформацією управління, що супроводжує їх. Мінімальний об'єм пам'яті модуля БПП повинен забезпечувати зберігання двох максимальних масивів даних з інформацією управління.

ППС орієнтована на обробку неперервного потоку даних за алгоритмами, які характеризуються модульністю і локальністю. Особливістю даної системи є можливість роботи в синхронному або асинхронному режимі. В синхронному режимі ППС синхронізується сигналами переривання, які проступають на процесорні модулі. Період поступлення цих сигналів рівний такту роботи системи  $T$ . В асинхронному режимі роботи відсутній спільний такт роботи ППС. Перехід  $i$ -ї сходинки системи від обробки попереднього масиву на обробку наступного масиву здійснюється після завершення  $i$ -ю сходинкою обробки попереднього масиву і завантаження в пам'ять  $i$ -го модуля БПП наступного масиву. Управління роботою ППС в асинхронному режимі здійснюється шляхом аналізу інформації управління, яка супроводжує масив даних, що передаються між сусідніми сходинками системи [15].

Кожен раз, коли  $i$ -та сходинка закінчує обробку вхідного масиву даних, результати обробки з інформацією управління розміщують в пам'яті  $(i+1)$ -го модуля БПП.

За допомогою інформації управління (i+1)-а сходинка ППС отримує інформацію про готовність до обробки нового масиву даних.

Основним параметром БПП є період звертання  $T_z$  до неї пристроїв, який залежить від кількості пристроїв і від часу  $t_{\text{ц}}$  циклу читання (запису) в ОЗП. Даний час для пристроїв з однаковою швидкістю виконання операцій вводу-виводу визначається виразом  $T_z \geq t_{\text{ц}} \times n$ .

Програма реалізується з використанням парадигми ООП і написана в середовищі Borland Delphi 7. Система Delphi дозволяє швидко та ефективно розробляти найрізноманітніші програми. Delphi є достатньо потужною системою візуального об'єктно-орієнтованого програмування. Дана система дає можливість створювати зручний користувацький інтерфейс, а широкий набір функцій, методів та властивостей дає можливість вирішувати прикладні розрахунково-обчислювальні задачі. В систему включено також зручні засоби відлагоджування. Традиційно Delphi відносять до RAD-систем (Rapid Application Development – швидка розробка програм). Це технологія візуального програмування, коли програміст оформляє свою майбутню програму і бачить результати своєї роботи ще до запуску програми.



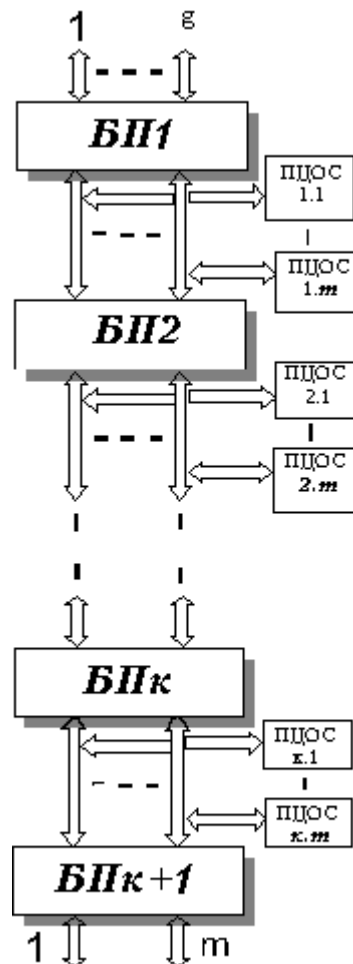


Рисунок 5.12. Структура паралельно-поточної системи з обміном через багатопортову пам'ять

Прикладні програми Delphi створюються в інтегрованому середовищі розробки (IDE – Integrated Development Environment). Користувацький інтерфейс даного середовища служить для взаємодії з програмістом і включає декілька вікон, які містять різні елементи керування. За допомогою засобів IDE розробнику зручно проектувати інтерфейсну частину програми, а також писати програмний код і зв'язувати його з елементами керування. В IDE проходять усі етапи створення програм, включаючи відлагодження.

Інтегроване середовище розробки Delphi 7 являє собою багатовіконну систему. Для будь-якої програми автоматично пропонується вікно (форма), для якої вже створено два файли – з описом і модулем. Таким чином створюється найпростіша програма. Найпростіша програма являє собою заготовку або каркас, який забезпечує розробника усім необхідним для програми. Так не

потрібно писати свій обробник подій клавіатури або драйвер мишки, а також створювати пакет процедур для роботи з вікнами. При розробці до найпростішої програми додаються нові форми, елементи керування, а також нові обробники подій.

Розробка програми в Delphi включає два взаємозв'язаних етапи: створення користувацького інтерфейсу і визначення функціональності програми. Користувацький інтерфейс програми визначає спосіб взаємодії користувача та програми, а саме, зовнішній вигляд форм при виконанні програми і те, яким чином користувач може керувати програмою. Інтерфейс розробляється шляхом розміщення у формі компонент, які називаються інтерфейсними компонентами або елементами управління. Функціональність програми визначається процедурами, які виконуються при виникненні певних подій, наприклад, тих, які відбуваються при діях користувача з елементами управління форми [20].

Розроблена програма призначена для реалізації прикладу використання БПП в ППС. Загальний вигляд програми представлений на рис. 5.13. Робота починається з введення вхідних даних, якими є кількість сходинок (тобто кількість БПП) в ППС, кількість ПЦОС в кожній сходинці, кількість вхідних інформаційних каналів, частота поступлення даних, величина масиву інформації та тривалість циклу читання або запису, та їх збереження кнопкою «Зберегти».

При візуалізації ППС передбачено, щоб на першій сходинці здійснювалось додавання двох масивів, а на другій - множення масиву на значення, яке вводиться користувачем. Тому передбачено введення двох масивів даних і їх збереження кнопкою «Зберегти масиви». Величина масивів даних відповідає величині масиву інформації, яка була введена раніше в вхідних даних. Моделювання роботи ППС здійснюється відповідною кнопкою - «Моделювання роботи ППС». Відображено кількість тактів, кількість сходинок, що відповідають певній кількості БПП, яка була введена в вхідних даних, самі дані, отримані результати кожної сходинки та інформація

управління До інформації управління належить код операції (в даному випадку це операція додавання з кодом ADD та операція множення з кодом MUL) та розмір масиву. Показана інформація, що міститься в кожному такті на кожній сходинці ППС. Максимальна кількість даних кожної сходинки та кількість отриманих результатів відповідає кількості ПЦОС кожної сходинки.

Також можна порахувати значення продуктивності кожного процесора, такт роботи системи, об'єм обчислень кожної під задачі та період звертання бо БПП.

Такт	БПі	Дані	Дані	Код	Розмір	Результат	Результат
1	БП1	10	12	Add	8		
	БП2						
	БП3						
2	БП1	14	16	Add	8		
	БП2	20	24	Mul	8		
	БП3						
3	БП1	18	20	Add	8		
	БП2	28	32	Mul	8		
	БП3	20	24	None	8		
4	БП1	22	24	Add	8		
	БП2	36	40	Mul	8		
	БП3	28	32	None	8	20	24
5	БП1						
	БП2	44	48	Mul	8		
	БП3	36	40	None	8	28	32
6	БП1						
	БП2						
	БП3	44	48	None	8	36	40
7	БП1						
	БП2						
	БП3					44	48

Вхідні дані

Кількість підзадач - K: 3

Кількість ПЦОС кожної сходинки - m: 2

Кількість вхідних інформаційних каналів - g: 2

Частота поступлення даних - F: 66

Величина масиву інформації - N: 8

Тривалість циклу читання/запису в ОЗП - Tц: 0,09

Продуктивність процесора: 8448

Такт роботи системи: 0,0606060606060606

Об'єм обчислень кожної підзадачі: 1024

Період звертання до БПП: 0,18

Зберегти Порахувати

Масиви даних

0	1	2	3	4	5	6	7
10	11	12	13	14	15	16	17

Зберегти масиви

Етап 1: Додавання двох масивів

Етап 2: Множення масиву на значення 2

Моделювання роботи ППС

Рисунок 5.13 – Загальний вигляд програми прикладу використання БПП в ППС.

## ВИСНОВКИ

1. У роботі запропоновані такі принципи побудови багато портової пам'яті: ієрархічності і багаторівневості побудови пам'яті великої ємності з широким використанням кешування і механізмів управління сигналами очікування, просторового і функціонального розділення внутрішньої пам'яті з широким використанням різних за швидкістю і шириною доступу модулів пам'яті, багатоканального доступу до пам'яті з метою забезпечення декількох звертань на протязі одного командного циклу, апаратної реалізації складних алгоритмів генерації послідовності адрес з використанням модульної арифметики, гармонічного поєднання можливостей інтегральної технології з розширенням функцій пам'яті, збільшення кількості внутрішніх і зовнішніх каналів доступу та з покращенням параметрів модулів пам'яті (ємності, швидкодії і т.д.).

2. Розроблено структуру багато портової пам'яті, основними елементами якої є: пристрій керування, контролери БПП та ОЗП.

3. Здійснено моделювання компонентів на VHDL, що дозволяє дослідити значення вхідних, вихідних і проміжних сигналів в часі.

4. Розроблена програмна реалізація прикладу використання багатопортової пам'яті в паралельно-потоківих системах в середовищі Delphi 7 компанії Borland.

5. Отримані результати свідчать про ефективність запропонованих підходів для розробки багатопортової пам'яті для комп'ютерних систем реального часу.

6. Подальші дослідження повинні проводитись в напрямку вдосконалення структурних рішень побудови багато портової пам'яті.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Угрюмов Е.П. Цифровая схемотехника: учеб. Пособие для вузов.- 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2010 - 816 .: ил
2. Метлицкий Е.А. Каверзнев В.В. Системы параллельной памяти: Теория проектирование, применение./Под ред В.И. Тимохина. Издательство Ленинградского университета. 1989 – 240 с.
3. Цмоць І.Г. Інформаційні технології та спеціалізовані засоби обробки сигналів і зображень у реальному часі. Львів: Монографія, 2005. – 730с.
4. AHDL - язык описания аппаратных средств: Учеб. пособие / А.С. Яицков; Под. ред. акад. В.С. Бурцева, акад. Б.С. Митина. М. : Изд-во МАТИ-РГТУ "ЛАТМЭС", 1998- 119с.
5. Угрюмов Е. П. Цифровая схемотехника. — СПб.: БХВ — Санкт-Петербург, 2000. — 528 с: ил.
6. Поляков А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры. – М.: СОЛОН-Пресс. 2003. – 320 с.: ил.
7. Параллельная обработка информации: в 5т./ АН УССР. Физ-мех. ин-т. – Киев: Наукова думка, 1984. – т5: Проблемно-ориентированные и специализированные средства обработки информации / А.И. Аксенов, В.В. Аристов, Е.Ю. Барзилович и др.; Под ред. Б.Н. Малиновского и Грицика В. В. – 1990. – 504с.
8. Шалыто А.А. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука, 2000. – 780 с.
9. Цмоць І.Г. Паралельно-потоківі системи цифрової обробки сигналів з обміном через багатопортову пам'ять // Вісник ДУ “Львівська політехніка”: "Комп'ютерні системи проектування. Теорія і практика ", №373. - Львів, 1999. - С.106-111.
10. Фельдман Л.П., Дедищев В.О. Математическое обеспечение САПР: Моделирование вычислительных и управляющих систем. – К.: УМК ВО, 1992. – 256 с.
11. Савельев А.Я. Арифметические и логические основы цифровых автоматов. М.: Высш. школа, 1980. – 225 с.

12. Голицына О.Л., Попов И.И. Основы алгоритмизации и программирования. - Издательство Форум, 2002. - 432с.
13. Решетняк В.Н., Гузик В.Ф., Сидоренко В.Г. «Проектирование распределенных информационно-вычислительных систем.» Учеб. пособие. Таганрог: ТРТУД 1996 год.
14. Карцев М.А., Брик В.А. Вычислительные системы и синхронная арифметика. - М.: Радио и связь, 1981. - 360с
15. Основы построения технических средств ЕС ЭВМ на интегральных микросхемах/В. В. Саморуков, В. М. Микитин, В. А. Павлычев и др./ Под ред. Б. Н. Файзулаева. — М.: Радио и связь, 1981. — 288 с.
16. Хоровиц. П., Хилл У. Искусство схемотехники: В 3-х томах: Т. 2: Пер с англ. — 4-е изд. — М.: Мир, 1993. — 371 с.
17. Бродин В. Б., Шагурин И. И. Микроконтроллеры: Справочник. — М.: ЭКОМ, 1999. - 395 с.
18. Кохонен Т. Ассоциативные запоминающие устройства: Пер. с англ. — М.: Мир, 1982, - 384 с.
19. Куприянов М. С, Матюшкин Б. Д. Цифровая обработка сигналов: Процессы. Алгоритмы. Средства проектирования. — СПб.: Политехника, 1998. - 592 с.
20. Лебедев О. Н. Мирошниченко А. И., Телец В. А. Изделия электронной техники. Цифровые микросхемы. Микросхемы памяти. Микросхемы ЦАП и АЦП: Справочник. — М.: Радио и связь, 1994. — 248 с.
21. Закон України “Про охорону праці”.
22. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. ДСанПіН 3.3.2.007 - 98.
23. В.Ц.Жидецький, В.С.Джигирей, В.М.Сторожук, Л.В.Туряб, Х.І.Лико., .
24. ГОСТ 12.1.003-83. ССБТ. Шум. Общин требования безопасности.
25. ГОСТ 12.1.004-91. ССБТ. Пожарная безопасность. Общие требования.

26. ГОСТ 12.1.005-88. ССБТ. Общин санитарно-гигиенические требования к воздуху рабочей зоны.
27. ГОСТ 12.1.006-86. ССБТ. Электромагнитные поля радиочастот. Допустимые уровни напряженности и требования к проведению контроля на рабочих местах.
28. ГОСТ 12.1.010-76 ССБТ. Взрывобезопасность. Общие требования.
29. ГОСТ 12.1.012-78. Вибрация. Общие требования безопасности.
30. ГОСТ 12.2.032-78. ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.
31. ГОСТ 12.2.033-78. ССБТ. Общие эргономические требования.
32. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних.
33. ДНАОП 0.00-1.31-99. Правила охорони праці при експлуатації ЕОМ. машин. ДСанПіН 3.3.2.007-98.
34. Правила устройства электроустановок (ПУЭ-85).
35. СНиП П-4-79. Природное и искусственное освещение. Нормы проектирования. - М. : Стройиздат, 1980. - 48с.
36. СНиП П-12-77. Защита от шума. - М. : Стройиздат, 1988, - 49с.