

УДК 056.55

Н.М. Ліщина

Луцький інститут розвитку людини Університету «Україна»

## МІСЦЕ ПАРАЛЕЛЬНОЇ ПАМ'ЯТІ З ВПОРЯДКОВАНИМ ДОСТУПОМ В АРХІТЕКТУРІ СУЧАСНИХ КОМП'ЮТЕРНИХ СИСТЕМ

*В статті оцінено місце паралельної пам'яті з впорядкованим доступом в архітектурі сучасних комп'ютерних систем. Розглянуто використання ПВД в процесорах векторних комп'ютерів в ролі векторних реєстрів. Описано як паралельна структура реконфігурованого прискорювача сприяє використанню паралельної архітектури СП на основі паралельної пам'яті.*

Ключові слова: паралельна пам'ять з впорядкованим доступом, процесори векторних комп'ютерів, спеціалізовані процесори.

**Постановка проблеми.** Пам'ять має визначальний вплив на технічні характеристики комп'ютера, особливо при створенні кластерних обчислювальних комплексів та професійних серверних систем. Тому існує значна потреба в покращенні технічних характеристик та функціональних можливостей пам'яті. Для цього, в першу чергу, потрібно забезпечувати паралельний доступ до даних в пам'яті та виконувати операції впорядкування даних в масивах, які зберігаються в пам'яті. Такі функції в повній мірі забезпечує лише пам'ять з впорядкованим доступом (ПВД), в якій здійснюється доступ до даних у програмно встановленому порядку, тобто індекс, який поступає в пам'ять разом з даним, або під час його зчитування, вказує місце даного у вихідному масиві. Були досліджені проблеми організації пам'яті, розроблені критерії порівняння різних типів пам'яті та показано, що ПВД є найефективнішою для роботи з масивами даних.

**Аналіз останніх досліджень і публікацій.** Питанням побудови пристроїв пам'яті комп'ютера присвячені роботи В. Корнійчука, А. Мельника, О. Палагіна, А. Шігіна, Ю. Шамаєва, Т. Кохонена, Д. Паттерсона та інших вчених.

Потрібно відзначити, що з розвитком комп'ютерів суттєві зміни відбулися і в будові пам'яті, значно покращилися її технічні характеристики. Разом з тим, як це було вище показано, при виконанні ресурсномістких задач потрібно забезпечувати зберігання масивів даних, що надходять з багатьох каналів, одночасно з зчитуванням раніше прийнятих масивів даних для опрацювання в багатоблокних операційних пристроях процесорів, виконувати операції реорганізації та впорядкування даних в масивах, що для існуючих типів пам'яті є доволі складною, а часто і не вирішуваною з прийнятними характеристиками задачею. Це зумовлено їх потенційними обмеженнями, які суттєво ускладнюють організацію роботи процесора та приводять до сповільнення його роботи. Тому існує значна потреба в покращенні технічних характеристик та функціональних можливостей пам'яті.

**Виклад основного матеріалу дослідження.** Сучасні комп'ютери мають архітектуру з спрощеною системою команд. Процесор цих комп'ютерів має конвеєрну структуру, коли засоби виконання ярусів потокового графа алгоритму виконання циклу команди розділяються конвеєрними реєстрами. Щоб підвищити продуктивність конвеєрного процесора потрібно далі спрощувати операції його ярусів та поглиблювати глибину конвеєра. Це і робиться в сучасних процесорах, в яких глибина конвеєра досягає двадцяти і більше ярусів. Наприклад, процесор комп'ютера UltraSPARC III має 10 ярусів конвеєра, а процесор комп'ютера Pentium IV – 20 ярусів конвеєра. Однак процес спрощення операцій ярусів конвеєра має межу, коли операції не піддаються поділу. Наприклад фаза вибірки команди з пам'яті не може бути поділеною на простіші фази. Тоді для підвищення продуктивності процесора необхідно використовувати паралельне включення декількох конвеєрів команд. Такі процесори з декількома конвеєрами команд дозволяють одночасно виконувати кілька скалярних команд і тому дістали назву суперскалярних.

Першу суперскалярну архітектуру мав процесор America, модифікований варіант якого під назвою POQER-1 (Performance Optimization Qith Enhanced RISC) впровадили до серійних систем RISC System/6000 фірми IBM. Нарешті, підмножину архітектури POQER-1 реалізовано в процесорах PoQer PC, які є основою комп'ютерів Apple Macintosh. Іншими прикладами суперскалярних процесорів є процесори систем UltraSparc фірми Sun та Alpha фірми DEC.

Структура суперскалярного процесора, та його зв'язки з кеш пам'яттю даних і команд, показані на рис.1. [1].



Рис.1. Структура суперскалярного процесора  
 Авторська розробка

Тут з пам'яті одночасно вибирається та декодується декілька команд, а блок виконання команд включає кілька функціональних блоків. Для забезпечення одночасного читання та запису кількох операндів кеш пам'ять будується за модульним принципом.

Зрозуміло, що підвищення продуктивності такого процесора досягається шляхом його конвеєризації. Діаграма виконання команд в суперскалярному процесорі, який має два конвеєри команд, показана на рис.2.

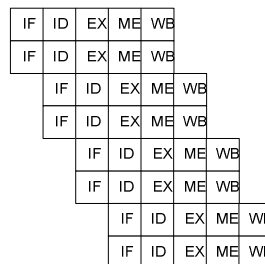


Рис. 2. Діаграма виконання команд в суперскалярному процесорі з двома конвеєрами команд  
 Авторська розробка

В скалярних та суперскалярних процесорах операції виконуються над скалярними даними [2]. Однак існує значна кількість задач, коли опрацюванню за одними процедурами підлягають великі масиви (вектори) даних. В цьому випадку виглядає доцільним розгляд можливості модифікації комп'ютера під виконання цього класу задач. До цих пір така модифікація здійснювалась в потужних комп'ютерах, але на даний час вона почала поширюватись на всі типи комп'ютерів. Відповідно комп'ютери, орієнтовані на опрацювання векторів даних, дістали назву векторних.

Різницю між виконанням скалярної та векторної операції наглядно відображає рис.3, з якого видно, що скалярна операція додавання передбачає дію над двома даними, тоді як векторна – над двома векторами даних.

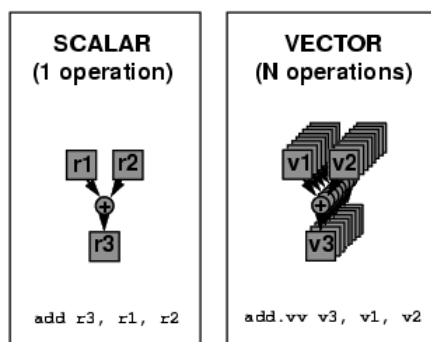


Рис.3. Виконання скалярної та векторної операцій додавання.  
 Авторська розробка

Наведемо приклад програми з скалярними і векторними кодами. Запишемо програму обчислення виразу  $Y = a * X + Y$ , де  $Y, X$  – вектори,  $a$  – скаляр. Нехай вектори мають довжину по 64 елементи. Векторна програма має вигляд [1]:

```
LD      F0, a          ; load scalar a
LV      V1, Rx         ; load vector X
MULTS   V2, F0, V1     ; vector-scalar mult.
LV      V3, Ry         ; load vector Y
ADDV    V4, V2, V3     ; add
SV      Ry, V4         ; store vector
```

Відповідна скалярна програма має вигляд:

```
LD      F0,a
ADDI    R4,Rx,#512    ; last address to load
Loop:   LD      F2,0(Rx) ; load X(l)
        MULTD F2,F0,F2  ; a*X(l)
        LD      F4, 0(Ry) ; load Y(l)
        ADDD   F4,F2,F4 ; a*X(l)+Y(l)
        SD     F4,0(Ry) ; store into Y(l)
        ADDI   Rx,Rx,#8 ; increment index
        ADDI   Ry,Ry,#8 ; increment index
        SUB    R20,R4,Rx ; compute bound
        BNZ   R20, loop ; check if done
```

У скалярній програмі курсивом позначено залежності, яких немає у векторному варіанті програми. Обидва варіанти програми можна порівняти за наступними кількісними характеристиками:

1. За кількістю операцій: 578(2+9\*64) проти 321(1+5\*64); кількість операцій у векторній програмі зменшено в 1.8 разів.
2. За кількістю команд: 578(2+9\*64) проти 6-ти команд у векторній програмі; перевага в 96 разів.

В табл. 1. наведені характеристики кількох промислових векторних комп'ютерів, з якої видно доцільність їх створення з огляду на досягнуту продуктивність [1].

Таблиця 1

Авторська розробка.

Тип машини	Рік випуску	Частота, МГц	Кількість регістрів	Кількість елементів	Кількість пристроїв float point	Кількість пристроїв load/store	Продуктивність (MFLOPS)
Cray-1	1976	80	8	64	6	1	160
Cray XMP	1983	120	8	64	8	2L, 1S	940
Cray YMP	1988	166	8	64	8	2L, 1S	2667
Cray C-90	1991	240	8	128	8	4	15238(16)
Cray T-90	1996	455	8	128	8	4	57600(32)
Conv. C-1	1984	10	8	128	4	1	20(1)
Conv. C-4	1994	133	16	128	3	1	3240(4)

Fuj. VP200	1982	133	8-256	32-1024	3	2	533(1)
Fuj. VP300	1996	100	8-256	32-1024	3	2	N/A
NEC SX/2	1984	160	8 + 8K	256 + var	16	8	1300(1)
NEC SX/3	1995	400	8 + 8K	256 + var	16	8	25600(4)

Таким чином, процесори векторних комп'ютерів виконують команди над векторами даних. За структурою це можуть бути процесори векторних комп'ютерів з складною та простою системою команд, конвеєрні та суперконвеєрні, а також процесори супервекторних комп'ютерів, коли в процесорі є декілька конвеєрів команд. Основна відмінність процесорів векторних комп'ютерів – забезпечення одночасного виконання однієї команди над вектором даних. Це, зокрема, дозволяє будувати їх блоки виконання команд за конвеєрним принципом і при цьому позбутися конфліктів, які суттєво гальмують роботу конвеєра чи ускладнюють його структуру [4].

Для виявлення базових принципів побудови процесорів векторних комп'ютерів розглянемо структуру та систему команд процесора векторного варіанту комп'ютера DLX, а саме комп'ютера DLXV [1,2]. До складу процесора, структура якого приведена на рис.3, входять пристрій векторного читання запису, регістрові файли з векторними та скалярними регістрами, а також операційний пристрій з набором конвеєрних операційних пристроїв додавання, множення та ділення з рухомою комою та виконання арифметичних та логічних операцій над цілими числами.

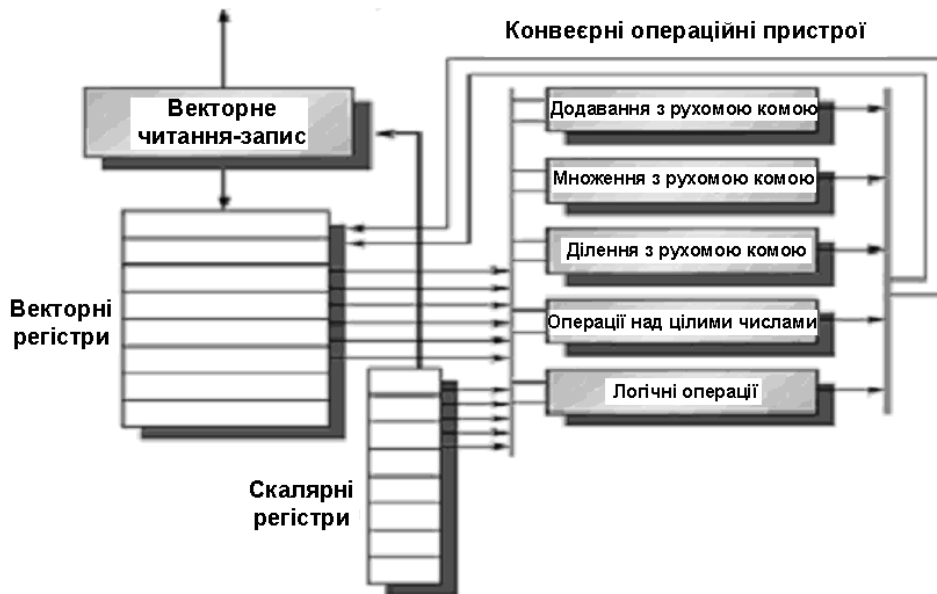


Рис.3. Структура процесора комп'ютера DLXV.  
 Авторська розробка

Цей комп'ютер має векторні команди, які приведено в табл.2.

Таблиця 2.

Авторська розробка

Команда	Операнди	Операція	Коментар
ADDV	V1, V2,V3	$V1=V2+V3$	VECTOR+VECTOR
ADDSV	V1, F0,V2	$V1=F0+V2$	SCALAR+VECTOR
MULTV	V1, V2,V3	$V1=V2 \times V3$	VECTORxVECTOR
MULSV	V1, F0,V2	$V1=F0 \times V2$	SCALAR x VECTOR
LV	V1, R1	$V1=M[R1..R1+63]$	LOAD, STRIDE=1
LVQS	V1, R1,R2	$V1=[R1..R1+63 \times R2]$	LOAD, STRIDE=R2
LVI	V1, R1,V2	$V1=[R1+V2i, i=0..63]$	indirect ("gather")
SeqV	VM,V1,V2	$V_{MASKi}=(V1i=V2i)$	comp. Set mask
MOV	VLR,R1	Vec. Len. Reg = R1	set vector length
MOV	VM, R1	Vec. Mask=R1	set vector mask

Наведений процесор є процесором комп'ютера з простою системою команд. До цього типу належать усі векторні суперкомп'ютери: Cray, Convex, Fujitsu, Hitachi, NEC.

Подібно до наведеного на рис.3, процесори векторних комп'ютерів містять наступні основні компоненти:

- Векторні реєстри - це реєстровий файл фіксованої ємності, що вміщує вектор даних. Цей файл має як мінімум 2 порти на читання і один порт на запис та зазвичай включає 8-32 векторних реєстрів, кожний з яких є 64-128-розрядним.
- Конвеєрні операційні пристрої. Зазвичай застосовують 4-8 операційних пристроїв, а саме додавання, множення та ділення з фіксованою та рухомою комою, зсуву, тощо.
- Векторний вузол читання-запису, також конвеєрний, який опрацьовує вектори даних. Водночас застосовують декілька таких вузлів.
- Скалярні реєстри, які містять один скаляр з рухомою комою або адресу.
- Багатощинні магістралі або комутаційні мережі, які з'єднують між собою всі зазначені компоненти аби прискорити роботу процесора в цілому.

Протягом останніх років векторні комп'ютери мали швидкий розвиток і з екзотичних перетворились в широкоживаний клас потужних комп'ютерів.

Спеціалізовані процесори (СП) створюють для реалізації вузького числа заданих алгоритмів. Вони займають значну частину ринку комп'ютерних засобів, тому питання дослідження принципів побудови СП постійно знаходиться в сфері уваги розробників комп'ютерів. На сьогодні важливість цього питання суттєво зросла. Це пояснюється появою реконфігурованих прискорювачів, які дозволяють суттєво підвищити продуктивність комп'ютерних систем. В них реалізуються СП, які виконують складні задачі опрацювання даних [1]. При цьому здатність до реконфігурування відриває нові можливості досягнення високих технічних характеристик комп'ютерної системи, до складу якої входить реконфігурований прискорювач. Більше того, паралельна структура реконфігурованого прискорювача сприяє використанню паралельної архітектури СП на основі паралельної пам'яті.

Це досягається шляхом орієнтації структури процесора на виконувани алгоритми апаратним способом. Такі процесори називають функціонально або апаратно-орієнтованими (АОП, в англійській літературі - ASIC - Application-Specific Integration Circuit).

Зараз, коли досягнення мікроелектронної технології підтримуються потужними САПР, створення апаратно-орієнтованих на виконувани алгоритми СП є реальною альтернативою універсальним процесорам. Такий підхід забезпечує максимально можливу продуктивність при вирішенні заданої задачі на кристалі заданого об'єму, вимагає мінімальних витрат обладнання на побудову СП для розв'язку заданої задачі за рахунок відшукування компромісу між програмними та апаратними засобами, характеризується простішим процесом проектування в порівнянні з універсальним процесором. Останньому сприяє високий рівень сучасних мов опису апаратних засобів. Тому процес проектування апаратно-орієнтованого процесора не є набагато складнішим навіть порівнюючи з розробкою спеціального програмного забезпечення для універсального процесора. Особливо ця перевага проявилась в даний час, коли сучасні САПР забезпечують повне відпрацювання моделей електронних компонент, включаючи їх роботу в складі СП, а також коли отримали широке розповсюдження технології повторного використання бібліотек раніше створених електронних компонент [1].

Зазвичай СП повинні обробляти інтенсивні потоки даних в темпі їх поступлення, тобто в реальному масштабі часу. Причому, одночасно з проведенням приймання вхідних даних має здійснюватись опрацювання раніше прийнятого масиву даних та зчитування масиву попередньо опрацьованих даних. У зв'язку з цим значне навантаження лягає на пам'ять СП, для побудови якої найкраще підходить ПВД [1]. На рис.5 наведено укрупнену узагальнену структуру СП у складі ПВД, арифметико-логічного пристрою та пристрою керування [3].

Арифметико-логічний пристрій СП виконує операції функціональних операторів виконуваних алгоритмів, для реалізації яких створюється СП. Пристрій керування формує інформацію, яка забезпечує доступ до даних в пам'яті, задає коди операцій арифметико-логічного пристрою, організує цикли та здійснює загальну синхронізацію.

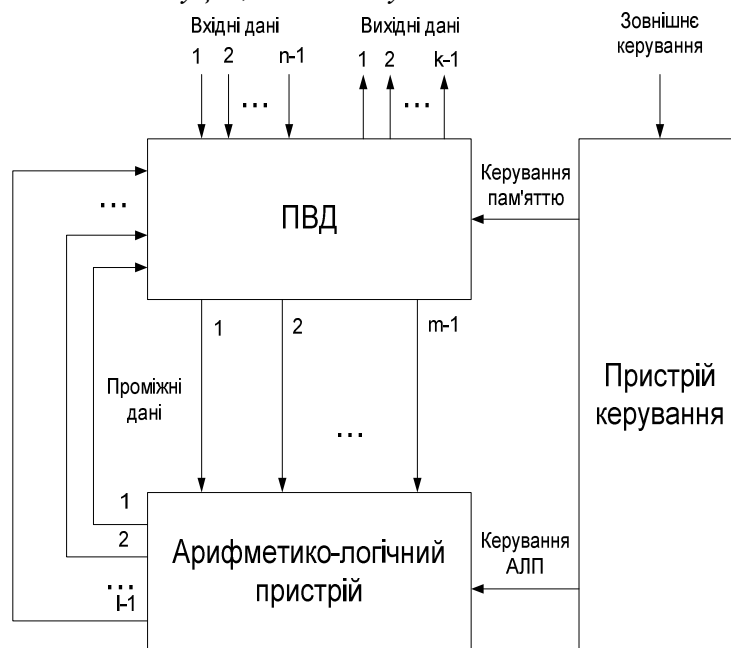


Рис.5. Узагальнена структура СП.  
Авторська розробка

Як видно з рис.5. ПВД приймає з  $n$  портів вхідні дані, видає на  $k$  портів вихідні дані, видає на  $m$  портів проміжні дані, які підлягають опрацюванню в арифметико-логічному пристрої, та запам'ятовує проміжні дані обчислень, які поступають з  $l$  портів арифметико-логічного пристрою.

#### Висновки

Таким чином, як видно з результатів аналізу місця та функцій пам'яті в комп'ютері, вона в значній мірі визначає його архітектурні особливості. Як витікає з проведеного аналізу, ПВД може бути використана в розглянутих процесорах векторних комп'ютерів в ролі векторних регістрів, що дозволить виключити потребу впорядкування даних після їх опрацювання в операційних пристроях. Більше того, виглядає доцільним її використання в ролі багатопортової пам'яті даних та в ролі багатопортової пам'яті команд. Паралельна структура реконфігуровного прискорювача сприяє використанню паралельної архітектури СП на основі паралельної пам'яті.

1. Мельник А.О. Архітектура комп'ютера / А.О. Мельник .- Луцьк: Волинська обласна друкарня, 2002. - 470 с.
2. Мельник А.О. Спеціалізовані комп'ютерні системи реального часу / А.О. Мельник .- Львів: ДУ "Львівська політехніка", 1996. – 54 с.
3. Мельник А.О. Порівняльний аналіз типів пам'яті комп'ютера / А.О. Мельник, Д.Х. Аль Раващдех // Вісник Національного університету "Львівська політехніка: комп'ютерні системи та мережі. - Львів : Львівська політехніка, 2007. - № 603.-с.81-86.
4. Мельник А.О. Програмовані процесори опрацювання сигналів / А.О. Мельник. - Львів.: ДУ „Львівська політехніка, 2000. - 57 с.