

## ЕФЕКТИВНІСТЬ ВІДКЛЮЧЕННЯ БЛОКІВ В ТАКТАХ ОСТАНОВА ПРИ ПРОЕКТУВАННІ ОБЧИСЛЮВАЧІВ

Інститут комп'ютерних технологій  
Національного авіаційного університету

*Наведено методику оцінки селективного відключення блоків з метою зниження енергоспоживання при проектуванні високопродуктивних обчислювачів*

### Актуальність теми

Одним з основних завдань логічного синтезу обчислювачів є розробка методів проектування із низьким енергоспоживанням. Актуальність цього визначається наявністю портативних комп'ютерів, засобів навігації, засобів зв'язку, цифрової аудіо- і відеотехніки, які повинні сполучати високу надійність і необхідну швидкість з низьким споживанням енергії для тривалої автономної роботи [1].

### Постановка задачі

Для вирішення питання про доцільність застосування методу зниження енергоспоживання необхідно визначити мо-

дель типового завдання, розробити систему моделювання, що дозволяє досліджувати режими роботи і завантаження основних функціональних вузлів обчислювача, що проектується. Результатом виконання цих завдань повинна бути методика оцінки ефективності застосування методу, яку можна використати при проектуванні будь-яких обчислювачів.

### Аналіз відомих досліджень

Відомі методи зниження енергоспоживання, які використовуються при проектуванні цифрових схем, мають свої достоїнства й недоліки. У табл. 1 проаналізовані методи зниження енергоспоживання.

Таблиця 1. Аналіз застосовності методів зниження енергоспоживання

| Метод   | Недоліки  | Можливість й доцільність застосування   |
|---|---|---|
| Зниження напруги живлення й зменшення проектних норм. | Потребує зменшення розмірів транзисторів.   | При відповідному рівні технології виготовлення.   |
| Динамічне керування частотою й напругою живлення.     | Для реалізації методу потрібна доробка програмного забезпечення й апаратних засобів.  | Використати в майбутньому при проектуванні обчислювальних систем зі знизеним енергоспоживанням.   |
| Введення в конвеєр додаткових тактів останова         | Приводить до зниження продуктивності. Час виконання програм стає непередбаченим, що виключає його застосування для систем реального часу. | Необхідні додаткові дослідження   |
| Використання декількох напруг в одному кристалі       | Вимагає одночасного використання декількох бібліотек стандартних осередків.   | На даному етапі не можливо використати через відсутність відповідних стандартних бібліотек елементів.   |
| Метод селективного відключення неживаних блоків       | Схема відключення займає площу кристала й розсіює енергію.  | Ефективність застосування методу визначається виконуваним завданням і трудозатратами на його реалізацію. Необхідно зробити аналіз доцільності використання для кожного режиму роботи. |

Результати проведеного аналізу дозволяють зробити висновок про перспективність дослідження методу селективного відключення невживаних блоків.

**Аналіз типового завдання**

Для визначення типового завдання, яке вирішується обчислювачем, проаналізовані реальні прикладні завдання, в яких використовується потокоорієнтований обчислювач. У табл. 2 показаний розподіл процесорного часу при виконанні основних типів операцій.

На основі аналізу типового завдання на базі потокоорієнтованого обчислювача побудована система для вирішення типового завдання (рис. 1). Ця система містить мікропроцесор, статичний ОЗП, ПЗП і контролер шини PCI, а також набір контролерів інтерфейсів, підключених до шини PCI. До складу типової системи вклю-

чено два канали інтерфейсу мультиплексного обміну (інтерфейс «манчестер» і магистральний паралельний інтерфейс Qbus). До складу системного контролера входять два послідовні канали RS232. До інтерфейсних шин підключені різні датчики і пристрої управління.

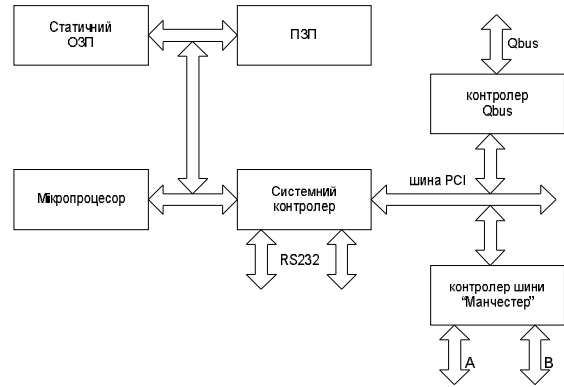


Рис. 1. Типова система

Таблиця 2. Приклади завдань для систем на базі потокоорієнтованого обчислювача

| Область застосування        | Використовувані інтерфейси          | Частка часу на цілочисельні обчислення | Частка часу для доступу до інтерфейсів | Частка часу на звернення до ОЗП (ПЗП) |
|-----------------------------|-------------------------------------|--|--|---------------------------------------|
| Навігаційна система         | Навігаційний контролер, 1553, RS232 | 50                                     | 20                                     | 30                                    |
| Управління рухомим об'єктом | 1553, RS232, Qbus                   | 35                                     | 35                                     | 30                                    |
| GPS-пристрої (картографія)  | 1553, відеовивід RS232              | 40                                     | 40                                     | 20                                    |
| Середнє                     |                                     | 41                                     | 32                                     | 27                                    |

**Дослідження структури енергоспоживання потокоорієнтованого обчислювача**

Для дослідження взято розроблений потокоорієнтований обчислювач [2]:

- конвеєр співпадає з «класичним» 5-ти стадійним конвеєром, описаним родоначальниками RISC-архітектури Паттерсоном і Хеннесі [3 - 4];
- відносна простота архітектури дозволяє ефективно використовувати автоматизовані засоби проектування, що значно прискорює процес розробки.

Структурна схема потокоорієнтованого обчислювача подана на рис. 2.

Розглянемо структуру конвеєра потокоорієнтованого обчислювача, яка подана на рис. 3.

Кожен такт 5-стадійного конвеєра розбитий на дві фази, в кожній з яких виконується певна дія. У табл. 3 вказані дії, які виконує обчислювач на кожній стадії і в кожній фазі конвеєра.

При нормальній роботі конвеєра в кожному такті можна починати виконання нової команди. Таким чином, виконання п'яти команд перекривається в часі (рис. 4). Такий конвеєр ефективний, оскільки різні ресурси процесора не потрібні декільком командам одночасно.

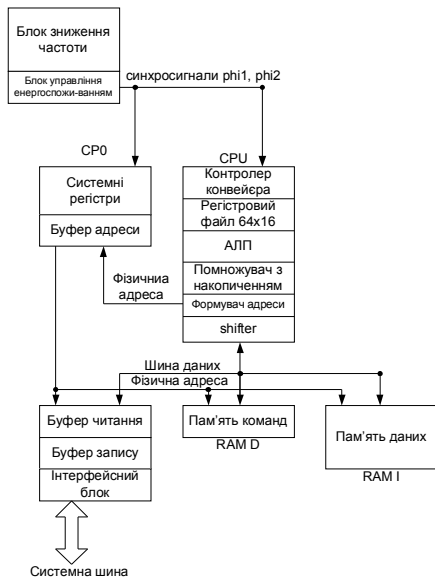


Рис. 2. Структурна схема потокоорієнтованого обчислювача

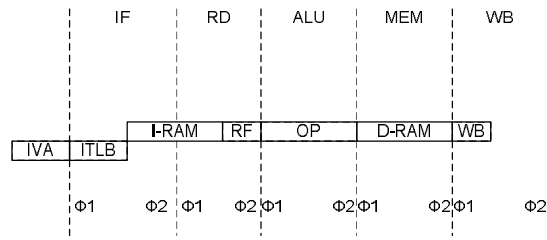


Рис. 3. Структура конвеєра потокоорієнтованого обчислювача

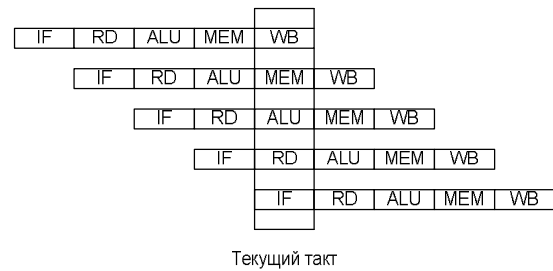


Рис. 4. Нормальна робота конвеєра

Таблиця 3. Розподіл операцій по стадіях конвеєра

|     |       |  |
|-----|-------|--|
| IF  | Φ1+Φ2 | Фізична адреса пересилається в RAM I.  |
| RD  | Φ1    | Обирається команда з RAM I (наявність або відсутність команди)   |
| RD  | Φ2    | Обирається операнд із регістрового файлу. Для команди умовного переходу обчислюється адреса переходу.                                  |
| ALU | Φ1+Φ2 | Обчислюються результати додавання, логічних операцій і т. п.   |
| ALU | Φ1+Φ2 | Для команди умовного переходу ухвалюється рішення про необхідність переходу. Для команд завантаження і збереження обчислюється адреса. |
| MEM | Φ1    | Для команд завантаження і запам'ятовування фізична адреса передається в RAM D.   |
| MEM | Φ2    | Для команд завантаження повертаються дані з RAM D.   |
| WB  | Φ1    | Запис результату в регістровий файл.   |

Такт складається з двох фаз (Φ1 і Φ2). Всі такти обчислювача можуть бути розбиті на два типи: робочі такти (*Run*), в цьому випадку обчислювач знаходиться в робочому режимі – відбувається просування конвеєра; холості такти або такти останову (*Stall*), такти, в яких не відбувається просування конвеєра.

Для потокоорієнтованого обчислювача можна виділити наступні причини переходу в режим останову:

**Останови, обумовлені читанням (*CM\_Stall – cache miss stall*).** Ініціюється, якщо відбулося непопадання в RAM I або RAM D. Тривалість останову залежить від характеристик зовнішньої пам'яті, до якої необхідно виконати звернення.

**Останов, обумовлений переповненням буфера запису (*WB\_Stall – Write Busy Stall*).** Якщо процесор намагається виконати команду збереження в той час, коли буфер запису переповнений (зайняті всі 4 входи), то процесор знаходитиметься в стані останову до тих пір, поки не закінчиться процедура запису на зовнішній шині і буфер запису не звільниться для прийому нової адреси і даних. Тривалість останову залежить від характеристик зовнішньої пам'яті, до якої необхідно провести звернення.

**Останов, обумовлений множенням з накопиченням (*HL\_Stall – HI/LO Busy Stall*).** Якщо програма намагається прочитати результат, що зберігається в регістрах блоку цілочисельного множення

з накопиченням (реєстри  $H1$  і  $LO$ ) в той час, коли виконується команда множення, ядро процесора перейде в стан останову до тих пір, поки не завершиться команда множення.

**Останов, обумовлений режимом Halt (Halt\_Stall).** Якщо в реєстрі *Cortfig* встановлений біт *Halt*, робота процесора зупиняється до появи зовнішнього запиту на переривання.

Аналіз можливих ситуацій, що приводять до виникнення тактів *stall*, показує, що у всіх цих тактах значна частина блоків обчислювача не виконує корисне

навантаження, тому неживані блоки є потенційними кандидатами на відключення під час тактів *stall*. Ці дані підтверджують доцільність застосування методу відключення неживаних блоків для обчислювача.

Результатом дослідження структури енергоспоживання обчислювача повинна бути таблиця, подібна табл. 4, на підставі якої можна зробити вибір режимів роботи обчислювача, для яких є передумови вважати, що застосування методу селективного відключення неживаних блоків приведе до зниження енергоспоживання.

Таблиця 4. Дані про активність блоків у вибраних режимах

| Режим роботи | Активні блоки           | Неактивні блоки           |
|--------------|-------------------------|---------------------------|
| Режим 1      | Перелік активних блоків | Перелік неактивних блоків |
| Режим 2      | Перелік активних блоків | Перелік неактивних блоків |
| .....        | Перелік активних блоків | Перелік неактивних блоків |
| Режим X      | Перелік активних блоків | Перелік неактивних блоків |

### Оцінка ефективності відключення блоків для типових режимів роботи обчислювача

Необхідно оцінити, наскільки ефективно виявиться відключення блоків для кожного з обраних режимів.

Нехай  $K_{mode} = P_{mode}/P_{total}$  – коефіцієнт, що показує ефективність відключення всіх неживаних блоків під час такту обраного режиму, де  $P_{mode}$  – потужність споживання обчислювача у обраному режимі, в припущенні, що неживані блоки відключені,  $P_{total}$  – потужність споживання обчислювача без відключення блоків.

$K_{mode}$  показує максимально можливу економію енергії для даного такту. Чим менше  $K_{mode}$ , тим більше економія.

Для знаходження  $K_{mode}$  використаємо базову формулу

$$P = V^2 f C_i, \quad (1)$$

де  $P$  – потужність системи,  $C_i$  – середнє значення ємності. Нехай схема, енергоспоживання якої оцінюється, складається з  $N$  блоків. Всі блоки працюють на одній частоті  $f$  і від однієї напруги живлення  $V$ . В цьому випадку потужність споживання  $P_{total}$  можна подати наступною формулою:

$$P_{total} = (C_{акт1} f + C_{акт2} f + \dots + C_{актN} f) V^2. \quad (2)$$

Для кожного типу останову із загальної кількості  $N$  можна виділити  $K$  блоків, які активні в даному такті. Вважати-мемо, що в тактах обраного режиму повністю відключено синхросигнал у всіх неактивних блоків, відповідно для цих блоків  $f=0$ . Тоді, потужність споживання в даному такті  $P_{mode}$ , в якому задіяні  $K$  блоків ( $K < N$ ) можна визначити за наступною формулою:

$$P_{mode} = (C_{вкл.блок1} f + C_{вкл.блок2} f + \dots + C_{вкл.блокK} f) V^2. \quad (3)$$

Коефіцієнт ефективності  $K_{mode}$  визначається таким чином:

$$K_{mode} = P_{mode}/P_{total} = (C_{блок1} + C_{блок2} + \dots + C_{блокK})/C_{total}. \quad (4)$$

Для знаходження  $K_{mode}$  потрібно знайти відношення ємностей активних блоків до неактивних. Пропонується замінити ємність значенням, що характеризує розмір блоку: кількістю транзисторів в блоці. При цьому передбачається, що всі елементи блоку перемикаються в кожному такті.

Знайдемо відношення ємностей активних блоків до загальної ємності схеми. Як приклад спеціального блоку узятий регістровий файл цілочисельного процесора. Даний блок має розмір масиву: 64 регістри по 16 розрядів кожен. Має один порт запису і два порти читання, що дозволяє виконувати три звернення до масиву за один такт. Розмір однієї запам'ятовуючої комірки – 8 транзисторів. Розмір масиву можна оцінити як 64 (кількість регістрів)\*16 (розрядність одного регістра) \*8 (розмір комірки в транзисторі)=8 тис. транзисторів.

Оскільки в одному такті можливе звернення до трьох регістрів, та кількість одночасно перемикаємих транзисторів оцінюється як 3 (кількість одночасно доступних комірок)\*16 (розрядність комірки) \*8 (розмір комірки в транзисторі)=0,4 тис. транзисторів.

Так само необхідно проаналізувати всі спеціальні блоки, що входять до складу обчислювача, що досліджується. Результатом такої роботи повинна бути таблиця, по структурі подібна табл. 5.

Таблиця 5. Розміри блоків з вказівкою кількості транзисторів, що одночасно перемикаються

| Назва блоку       | Кіл-ть транзисторів в блоці (тис. транз.) $S$ | Кіл-ть одночасно перемикаємих транзисторів (тис. транз.) $S'$ |
|-------------------|---|---|
| Обчислювач цілком | $S_{total}$                                   | $S'_{total}$  |
| Блок 1            | $S_{блок1}$                                   | $S'_{блок1}$  |
| Блок 2            | $S_{блок2}$                                   | $S'_{блок2}$  |
| .....             | .....   | .....   |
| Блок $N$          | $S_{блокN}$                                   | $S'_{блокN}$  |

Враховуючи дані табл. 5, формулу (4) можна подати в наступному вигляді:

$$K_{mode} = (S'_{блок1} + S'_{блок2} + \dots + S'_{блокN}) / S'_{total}. \quad (5)$$

Отримана формула дозволяє оцінити ефективність відключення неживаних блоків в тактах обраного режиму. Для

цього необхідно підставити у формулу (5) дані з табл. 4 і дані з табл. 5.

Результатом повинна бути таблиця з коефіцієнтами ефективності відключення неживаних блоків для обраних режимів роботи (табл. 6).

Таблиця 6. Коефіцієнти ефективності відключення неживаних блоків в тактах останову

| Тип циклу останову | Коефіцієнт ефективності $K_{mode}$ |
|--------------------|------------------------------------|
| Режим 1            | $K_{mode1}$                        |
| Режим 2            | $K_{mode2}$                        |
| ....               | ....                               |
| Режим $X$          | $K_{modeX}$                        |

### Метод оцінки ефективності селективного відключення блоків обчислювача

Опишемо метод оцінки ефективності селективного відключення блоків обчислювача з погляду економії енергоспоживання.

Метод може бути подана у вигляді наступної послідовності дій:

1. Аналіз області застосування. Визначення і аналіз типового завдання.

На основі технічних вимог, що пред'являються до обчислювальної системи (ОС), що проектується визначити типове завдання. Провести його аналіз для визначення тимчасового співвідношення основних типів операцій, таких, як цілочисельні обчислення, звернення до пам'яті і інтерфейсних пристроїв.

*Початкові дані:* Технічні вимоги до ОС.

*Результат:* Таблиця з розподілом часу виконання основних типів операцій.

2. *Синтез моделі типового завдання.*

Для дослідження поведінки блоків обчислювача побудувати модель типового завдання.

*Початкові дані:* Розподіл часу виконання основних типів операцій для типового завдання.

*Результат:* Модель типового завдання.

3. *Визначення структури типової системи.*

Для дослідження поведінки блоків обчислювача побудувати типову ОС.

*Початкові дані:* Технічні вимоги до ОС, типове завдання.

*Результат:* Структура типової системи.

4. *Вибір прототипного обчислювача.*

Є доцільним при проектуванні узяти за основу структуру одного з розроблених і допрацювати її з урахуванням технічних вимог і наявних обмежень.

*Початкові дані:* технічні вимоги до ОС; типове завдання; технологічні і тимчасові обмеження, що накладаються на розробника.

*Результат:* структура прототипного обчислювача.

5. *Дослідження режимів роботи прототипного обчислювача.*

Оцінити алгоритм роботи обчислювача, що розробляється, і відповідно, розглянути режими його роботи з погляду використання.

*Початкові дані:* структура прототипного обчислювача; опис режимів роботи прототипного обчислювача.

*Результат:* таблиця типових режимів роботи з вказівкою переліку блоків, що використовуються і неживаних блоків для кожного режиму.

6. *Визначення відношення ефективних розмірів блоків обчислювача до розміру всього обчислювача.*

Визначити розміри блоків обчислювача (кількість активних елементів блоку). На основі алгоритму роботи обчислювача встановити коефіцієнти вірогідності перемикання для кожного блоку. Визначити ефективні розміри кожного блоку.

*Початкові дані:* структура прототипного обчислювача; опис режимів роботи прототипного обчислювача.

*Результат:* Таблиця з ефективними розмірами кожного блоку.

7. *Розрахунок коефіцієнтів ефективності відключення неживаних блоків для кожного із заданих режимів роботи.*

Для кожного з режимів, визначених в п.5 методу, використовуючи ефективні розміри блоків, розрахований в п.6 метод, розрахувати коефіцієнти ефективності відключення неживаних блоків  $K_{mode}$  за формулою (4).

*Початкові дані:* перелік типових режимів роботи; перелік використовуваних і неживаних блоків для кожного режиму; таблиця з ефективними розмірами кожного блоку.

*Результат:* таблиця коефіцієнтів ефективності відключення неживаних блоків для кожного із заданих режимів.

8. *Аналіз результатів. Ухвалення рішень по реалізації схем відключення неживаних блоків.*

На основі таблиці коефіцієнтів ефективності, враховуючи вимоги по швидкодії і енергоспоживанню, а також доступні тимчасові, трудові і фінансові ресурси прийняти рішення по реалізації схем відключення неживаних блоків.

*Початкові дані:* таблиця коефіцієнтів ефективності відключення блоків для обраних режимів; технічні вимоги; тимчасові і фінансові обмеження.

*Результат:* Перелік ухвалених рішень по реалізації схем відключення неживаних блоків.

Метод у вигляді блок-схеми подана на рис. 5. Даний метод застосовано для порівняльного дослідження і підвищення ефективності селективного відключення

невживаних блоків для обчислювачів будь-яких областей застосування.

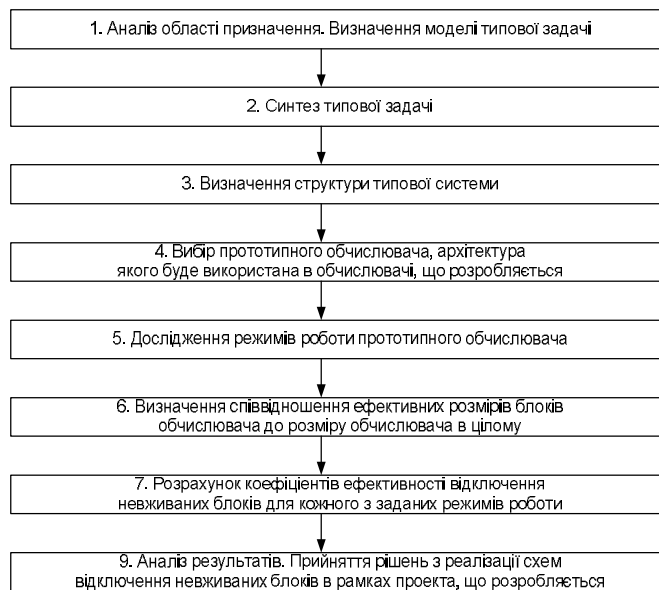


Рис. 5. Блок-схема методу оцінки ефективності відключення блоків обчислювача

### **Застосування розробленого методу при проектуванні потокоорієнтованого обчислювача**

Відповідно до першого пункту методу оцінки ефективності селективного відключення блоків потрібно провести аналіз області застосування обчислювача, що розробляється, визначення і синтез типового завдання.

Відповідно до технічних вимог, потокоорієнтований обчислювач призначався для застосувань, що вимагають високої швидкодії.

Відповідно до другого пункту методу, потрібно було побудувати модель типового завдання. Результатом даного пункту методу є модель типового завдання: 41% часу періоду займається обробкою цілочисельних даних, 59% – обмінами із зовнішніми пристроями і доступом до оперативної пам'яті.

Відповідно до третього пункту методу, потрібно було визначити типову систему. Результатом даного пункту методики є структура типової системи, яка подана на рис. 1.

У відповідності з четвертим пунктом методу, потрібно було обрати прототип обчислювача і розробити структуру

обчислювача. Результатом даного пункту методики є структура прототипу (рис. 2).

Відповідно до п'ятого пункту методу потрібно було досліджувати режими роботи обчислювача, що розробляється, з погляду використання. Проведено розгляд різних режимів роботи обчислювача. В результаті аналізу алгоритму функціонування обчислювача для обраного типового завдання виділено чотири режими останову, для яких застосування методу селективного відключення неживаних блоків потенційно може дати істотний ефект. Результатом даного пункту є табл. 7, побудована за шаблоном табл. 4.

Відповідно до п.6 методу потрібно розрахувати відношення ефективних розмірів блоків до розміру всього обчислювача. Як результат виконання п. 6 методики, використовуючи табл. 5 як шаблон, була складена табл. 8, де подано перелік всіх блоків потокоорієнтованого обчислювача з вказівкою кількості транзисторів в блоці та кількість одночасно перемикаємих транзисторів для всіх блоків.

Відповідно до п.7 методу необхідно розрахувати коефіцієнти ефективності відключення неживаних блоків для кожного з обраних режимів роботи. Коефіцієнти розраховуються за формулою (5) на

підставі даних табл. 7. Результатом пункту 7 методу є табл. 9 з коефіцієнтами ефе-

ктивності для кожного з режимів, побудована за шаблоном табл. 6.

Таблиця 7. Дані про активність блоків в режимах останову на прикладі потокоорієнтованого обчислювача

| Режим останову    | Активні блоки  | Неактивні блоки   |
|-------------------|--|---|
| <i>CM_Stall</i>   | Блок зниження частоти, буфер читання і контролер шини інтерфейсного блоку.       | <i>RAM I, RAM D</i> , цілочисельний процесор, системний співпроцесор, буфер запису інтерфейсного блоку.                         |
| <i>WB_Stall</i>   | Блок зниження частоти, буфер запису і контролер шини інтерфейсного блоку.        | <i>RAM I, RAM D</i> , цілочисельний процесор, системний співпроцесор, буфер читання інтерфейсного блоку.                        |
| <i>ILL_Stall</i>  | Блок зниження частоти, пристрій множення з накопиченням цілочисельного процесора | <i>RAM I, RAM D</i> , цілочисельний процесор окрім пристрою множення з накопиченням, системний співпроцесор, інтерфейсний блок. |
| <i>Halt_Stall</i> | Блок зниження частоти  | <i>RAM I, RAM D</i> , цілочисельний процесор, системний співпроцесор, інтерфейсний блок.  |
| Режим останову    | Активні блоки  | Неактивні блоки   |
| <i>CM_Stall</i>   | Блок зниження частоти, буфер читання і контролер шини інтерфейсного блоку.       | <i>RAM I, RAM D</i> , цілочисельний процесор, системний співпроцесор, буфер запису інтерфейсного блоку.                         |
| <i>WB_Stall</i>   | Блок зниження частоти, буфер запису і контролер шини інтерфейсного блоку.        | <i>RAM I, RAM D</i> , цілочисельний процесор, системний співпроцесор, буфер читання інтерфейсного блоку.                        |
| <i>ILL_Stall</i>  | Блок зниження частоти, пристрій множення з накопиченням цілочисельного процесора | <i>RAM I, RAM D</i> , цілочисельний процесор окрім пристрою множення з накопиченням, системний співпроцесор, інтерфейсний блок. |
| <i>Halt_Stall</i> | Блок зниження частоти  | <i>RAM I, RAM D</i> , цілочисельний процесор, системний співпроцесор, інтерфейсний блок.  |

Таблиця 8 Розміри блоків потокоорієнтованого обчислювача

| Назва блоку  | Кіл-ть транзисторів у блоці (тис. транз.) <i>S</i> | Кіл-ть одночасно перемикаємих транзисторів (тис. транз.) <i>S'</i> |
|--|--|--|
| Обчислювач цілком                                  | <b>119</b>   | <b>95,6</b>  |
| Цілочисельний процесор ( <i>CPU</i> )              | <b>73</b>  | <b>68</b>  |
| Контролер конвеєра                                 | 15   | 15   |
| Регістровий файл 64*16                             | 8  | 3  |
| АЛП  | 15   | 15   |
| Пристрій множення з накопиченням                   | 30   | 30   |
| Формувач адреси                                    | 5  | 5  |
| Системний співпроцесор управління ( <i>CPO</i> )   | <b>6</b>   | <b>6</b>   |
| Контрольні регістри і регістри управління пам'яттю | 3  | 3  |
| Буфер адреси ( <i>TLB</i> )                        | 3  | 3  |
| Інтерфейсний блок                                  | <b>15</b>  | <b>15</b>  |
| Буфер читання                                      | 5  | 5  |
| Буфер запису                                       | 5  | 5  |



|                       |    |     |
|-----------------------|----|-----|
| Контролер шини        | 5  | 5   |
| RAM даних             | 10 | 0,8 |
| RAM команд            | 10 | 0,8 |
| Блок зниження частоти | 5  | 5   |

Таблиця 9. Коефіцієнти ефективності відключення неактивних блоків в тактах останову

| Тип циклу останову | Коефіцієнт ефективності | Виграш в енергоспоживанні (%) |
|--------------------|-------------------------|-------------------------------|
| <i>CM_Stall</i>    | 0,15                    | 85                            |
| <i>WB_Stall</i>    | 0,15                    | 85                            |
| <i>HL_Stall</i>    | 0,36                    | 64                            |
| <i>Halt_Stall</i>  | 0,05                    | 95                            |

Як видно з табл. 9, якщо розглядати один такт знаходження обчислювача в режимі останову, метод відключення неживаних блоків дає істотний виграш в енергоспоживанні для кожного з розглянутих режимів роботи.

### Висновки

1. Проведено аналіз типового завдання. Складено тимчасовий розподіл основних типів операцій, на підставі чого синтезована модель типового завдання.

2. На базі потокоорієнтованого обчислювача розроблена структура обчислювальної системи, яка дозволяє виконувати типове завдання.

3. Проведено дослідження структури і режимів роботи типового потокоорієнтованого обчислювача. Запропоновані формули і типові форми подання даних, що дозволяють розрахувати коефіцієнти ефективності відключення неживаних блоків в тактах останову для заданих режимів.

4. Запропоновано метод оцінки ефективності селективного відключення блоків.

В результаті застосування методу оцінки ефективності селективного відключення блоків обчислювача при проектуванні досягнуті наступні результати:

1) на підставі аналізу даних, отриманих від потенційних користувачів потокоорієнтованого обчислювача, що розробляється, визначено типове завдання;

2) визначена структура типової системи. Досліджені режими роботи потоко-

орієнтованого обчислювача в системі. Виділено чотири режими останову, для яких селективне відключення неживаних блоків потенційно може дати істотний ефект. Для кожного режиму виділені активні і неактивні блоки;

3) обчислені коефіцієнти ефективності відключення неживаних блоків при виконанні моделі типового завдання для кожного з обраних режимів роботи. На підставі отриманих результатів зроблено висновок про доцільність реалізації схеми відключення неживаних блоків для режимів останову.

Застосування методу дозволить знизити ризик ухвалення помилкових рішень на ранній стадії розробки, за рахунок чого понизити трудомісткість і загальний час проектування.

### Список літератури

1. Красовська Є.В. Зменшення розсіяної потужності при проектуванні електронних пристроїв на основі КМОН-технології // Проблеми інформатизації та управління: Зб. наук. пр. – К.: НАУ, 2009. – Вип. 4 (28). – С. 81–87.

2. Красовська Є.В., Єфимець В.М. Високопродуктивний потокоорієнтований обчислювач // Проблеми інформатизації та управління: Зб. наук. пр. – К.: НАУ, 2010. – Вип. 1 (29). – С. 105–110.

3. Patterson D., Hennesy J. Computer Architecture and Quantitative Approach // Morgan Kaufmann Publishers, Inc., San Francisco, California, 1996.

Подано до редакції 19.04.10