

УДК 004.382, 004.383

В.А. Мельник

Національний університет «Львівська політехніка»

АНАЛІЗ ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ АПАРАТНИХ ПРИСКОРЮВАЧІВ ДЛЯ ПОБУДОВИ РЕКОНФІГУРОВНИХ КОМП'ЮТЕРНИХ СИСТЕМ

Мельник В.А. Аналіз та дослідження ефективності застосування апаратних прискорювачів для побудови реконфігурованих комп'ютерних систем. В статті проведено порівняльний аналіз типів архітектури реконфігурованих комп'ютерних систем і досліджено ступінь впливу обчислювальної складності задачі на величину прискорення, що може бути отримане в комп'ютерній системі шляхом під'єднання до неї пристроїв реконфігурованої логіки залежно від способу їх під'єднання, а також запропоновано методику визначення доцільності застосування реконфігурованої логіки з позиції організації зв'язку між нею і комп'ютером.

Ключові слова: реконфігуровані комп'ютерні системи, реконфігурована логіка, реконфігурований прискорювач, архітектура комп'ютерних систем.

Форм. 5. Табл. 0. Рис. 5. Літ. 23.

Melnyk V.A. Analysis and study of hardware accelerators application efficiency for reconfigurable computer systems design. The paper provides comparative analysis of the reconfigurable computer systems architecture types and investigates degree of influence of computing complexity to the value of performance that can be achieved in the computer system with connection of reconfigurable logic depending on the method of connection. The method of determination of the reconfigurable logic application expediency regarding its interfacing with computer is proposed.

Key words: reconfigurable computer systems, reconfigurable logic, reconfigurable accelerator, computer systems architecture.

Equat. 5. Tabl. 0. Fig. 5. Ref. 23.

Мельник В.А. Анализ и исследования эффективности применения аппаратных ускорителей для построения реконфигурируемых компьютерных систем. В статье проведен сравнительный анализ типов архитектуры реконфигурируемых компьютерных систем и исследована степень влияния вычислительной сложности задачи на величину ускорения, которая может быть получена в компьютерной системе путем подключения к ней устройств реконфигурируемой логики в зависимости от способа их подключения, а также предложена методика определения целесообразности применения реконфигурируемой логики с позиции организации связи между ней и компьютером.

Форм. 5. Табл. 0. Рис. 5. Лит. 23.

Вступ

Постановка проблеми. Реконфігуровані комп'ютерні системи (РККС) набули сьогодні значного поширення, що пов'язано з можливістю досягнення ними високих показників продуктивності шляхом апаратного прискорення виконання обчислювальних завдань в кристалах реконфігурованої логіки, що входять до їх складу. РККС склали конкуренцію іншим типам високопродуктивних комп'ютерних систем завдяки високим технічним характеристикам сучасних кристалів програмовних логічних інтегральних схем (ПЛІС) – апаратній основі реконфігурованого середовища РККС, та досягненням у галузі технологій проектування спеціалізованих процесорів, що підлягають синтезу в реконфігурованому середовищі РККС.

Підключення синтезованих в реконфігурованому середовищі спеціалізованих процесорів, структура яких в тій чи іншій мірі враховує особливості виконуваних алгоритмів, до комп'ютерної системи на основі універсальних процесорів, дозволяє до 2-3 порядків підняти її продуктивність. Можливість реконфігурування та синтезу в реконфігурованому середовищі спеціалізованих процесорів з новою структурою та функціями дозволяє змінювати функціональну орієнтацію створеної таким чином РККС зі збереженням її високої продуктивності на нових класах задач.

Сьогодні існує велика кількість практичних реалізацій та декілька варіантів архітектури реконфігурованих комп'ютерних систем, які відрізняються методами під'єднання пристроїв реконфігурованої логіки та характеристиками використовуваних інтерфейсів.

Аналіз останніх досліджень і публікацій. Реконфігуровані комп'ютерні системи, їх будову та організацію функціонування на різних рівнях висвітлено в роботі [19]. В роботі [1] розглянуто питання створення високопродуктивної РККС на основі персонального комп'ютера та реконфігурованого прискорювача, побудованого на основі ПЛІС. В роботі [4] висвітлено типи архітектури РККС, сформульовано спосіб виконання обчислень в цих системах, а також визначено проблемні питання їх застосування та окреслено шляхи вирішення цих питань.

Невирішені частини проблеми. Поряд із значним обсягом літератури, яка висвітлює різні аспекти будови, проектування, застосування РККС, описує реалізацію різноманітних прикладних алгоритмів в РККС, сьогодні відсутні літературні джерела, в яких проведено ґрунтовний і цілісний порівняльний аналіз типів архітектури РККС та досліджено вплив обчислювальної складності задачі на величину прискорення, що може бути отримане в комп'ютерній системі шляхом під'єднання до неї пристроїв реконфігурованої логіки. Це в значній мірі стримує розвиток напрямку реконфігурованих обчислень в цілому.

Метою дослідження є проведення порівняльного аналізу типів архітектури РККС і

дослідження ступеня впливу обчислювальної складності задачі на величину прискорення, що може бути отримане в комп'ютерній системі шляхом під'єднання до неї пристроїв реконфігуровної логіки залежно від способу їх під'єднання, а також розроблення методики визначення доцільності застосування реконфігуровної логіки з позиції організації зв'язку між нею і комп'ютером.

1. Дослідження та порівняльний аналіз типів архітектури реконфігуровних комп'ютерних систем

Тривалість t_{exe}^P виконання програми відповідно до описаного в роботі [4] способу опрацювання інформації можна подати таким виразом:

$$t_{exe}^P = T_{UPP}(P_{UPP}) + T_{SP}(P_{RE}) - T_{UPP,SP}^{PAR} + ttr_{data}, \quad (1)$$

де: $T_{UPP}(P_{UPP})$ – тривалість виконання підпрограми P_{UPP} універсальним процесором UPP ; $T_{SP}(P_{RE})$ – тривалість виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP , синтезованим в реконфігуровному середовищі РККС; ttr_{data} – тривалість пересилання даних між спеціалізованим процесором SP та універсальним процесором UPP чи пам'яттю під час виконання програми P_{in} ; $T_{UPP,SP}^{PAR}$ – час, на протязі якого універсальний процесор UPP та спеціалізований процесор SP виконують обчислення паралельно.

З виразу (1) випливає, що суттєвий вплив на значення тривалості t_{exe}^P виконання програми в РККС має її архітектура. По-перше, від типу використовуваного комп'ютера залежить тривалість $T_{UPP}(P_{UPP})$ виконання підпрограми P_{UPP} універсальним процесором UPP . По-друге, від технічних характеристик реконфігуровного середовища на основі ПЛІС та архітектури синтезованого в ньому спеціалізованого процесора SP залежить тривалість $T_{SP}(P_{RE})$ виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP . По-третє, суттєвий вплив має якість виконаного користувачем розподілу навантаження між універсальним та спеціалізованим процесорами, в тому числі і на час $T_{UPP,SP}^{PAR}$, на протязі якого універсальний процесор UPP та спеціалізований процесор SP виконують обчислення паралельно. І по-четверте, від типу архітектури залежить значення тривалості пересилання даних між універсальним комп'ютером та реконфігуровним середовищем. Тому актуальним завданням є дослідження архітектури РККС та проведення порівняльного аналізу її типів.

Відповідно до описаних в роботі [4] базових структур комп'ютерних систем з реконфігуровною логікою, розглянемо відповідні їм типи архітектури РККС, а саме:

- архітектура РККС з слабозв'язаною реконфігуровною логікою;
- архітектура РККС з тіснозв'язаною реконфігуровною логікою, підключеною до шини пам'яті;
- архітектура РККС з тіснозв'язаною реконфігуровною логікою, підключеною до шини процесора;
- архітектура РККС з інтегрованою в АЛП універсального процесора реконфігуровною логікою;
- архітектура РККС з інтегрованими в реконфігуровну логіку універсальним і спеціалізованим процесорами.

РККС з інтегрованою реконфігуровною логікою характеризуються найвищою швидкістю взаємодії універсального процесора з спеціалізованим, оскільки ці процесори фізично розміщені в одному кристалі, а це дає можливість реалізувати між ними такі зв'язки, які забезпечать максимальну швидкість виконання обчислювального алгоритму. При цьому вища швидкість взаємодії досягається в тих РККС, в яких спеціалізований процесор використовується як один з операційних пристроїв універсального процесора, розширюючи його систему команд. Дещо нижчою є швидкість взаємодії в РККС, де універсальний процесор є вбудованим в реконфігуровне середовище і виконує функції керування спеціалізованими процесорами. Характеристики продуктивності такого процесора є невисокими, оскільки він повинен займати мінімум площі кристалу, а обмін інформацією між ним і спеціалізованими процесорами здійснюється через внутрішньокристалеву системну шину. Для виходу на цю шину універсальний процесор

обладнано відповідним інтерфейсом, причому, якщо процесор є «жорстко» реалізованим в кристалі ПЛІС, тобто як замовний інтегральний елемент, так само «жорстко» реалізованою є й внутрішньокристална системна шина. Цим самим інтерфейсом обладнують і спеціалізовані процесори, що здійснюють на етапі їх проектування. По суті, цей підхід є реалізацією «комп'ютерної системи на програмовному кристалі» (англ. *SoPC – System on a Programmable Chip*). Зусиллями ряду фірм сьогодні створено відповідні інтерфейси, найпоширенішими з яких є *AMBA* [6] та *Wishbone* [23].

Принциповим недоліком архітектури РККС з інтегрованою реконфігуровною логікою є обмеження її ресурсів ємністю кристалу. Для нарощення ресурсів та підвищення продуктивності, а також надання РККС властивості масштабованості, до універсального процесора підключають зовнішню реконфігуровну логіку, що, власне, і є реалізацією тіснозв'язаної та слабозв'язаної типів архітектури. Зрозуміло, що швидкість взаємодії універсального процесора з спеціалізованим тут є значно нижчою.

Пристрої, на яких розміщують зовнішню по відношенню до універсального процесора реконфігуровну логіку, отримали назву реконфігуровних прискорювачів, оскільки їх можна під'єднати до довільного універсального комп'ютера, у тім числі й персонального, обладнаного сумісним інтерфейсом, з метою прискорення виконання ним обчислювальних завдань. Універсальний комп'ютер, до якого під'єднують реконфігуровний прискорювач, називають хост-комп'ютером.

Сучасний реконфігуровний прискорювач будують у вигляді однієї або декількох плат з розміщеними на них одним або декількома кристалами ПЛІС та одним або кількома роз'ємами для підключення зовнішніх пристроїв (плат розширення). Плати розширення підключають до базових плат через ці роз'єми. До плат розширення відносяться: плати, які підключаються до локальної шини базової плати, модулі *DIME (DSP and Image processing Module for Enhanced FPGAs)* [10] та плати введення-виведення (включаючи *WSDP*, волоконно-оптичні *Quad GLink*, *FPDP* і *A/D* плати).

2. Організація функціонування РККС з слабозв'язаною реконфігуровною логікою

В даній архітектурі прискорювач підключається до периферійної шини комп'ютера, зазвичай *PCI* чи *PCI Express*, *Ethernet* або *USB*. Застосування цієї архітектури вимагає оптимізації потоків вхідних та вихідних даних і є ефективним при їх малих об'ємах, оскільки шина легко перенавантажується, і пересилання даних до та з прискорювача може зайняти більше часу, ніж виконання алгоритму на центральному процесорі. Цей підхід є, проте, відносно дешевим.

Загальну схему РККС на основі універсального комп'ютера з слабозв'язаним прискорювачем зображено на рис. 1.

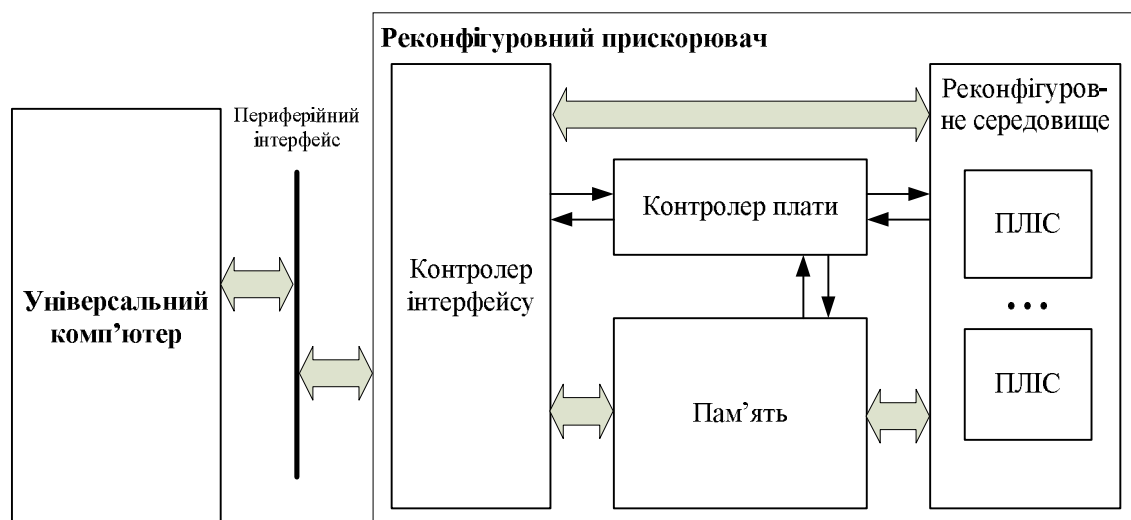


Рис. 1. Загальна схема РККС на основі універсального комп'ютера з слабозв'язаним прискорювачем

В архітектурі РККС на основі універсальних процесорів з слабозв'язаними прискорювачами в якості прискорювачів використовуються:

- змінні плати універсального комп'ютера, наприклад *ClearSpeed Advance X620* і *e620 Accelerator Boards* [8], або *NALLATECH H100 Series* [14], які взаємодіють з універсальним комп'ютером через один з периферійних інтерфейсів, найчастіше *PCI-Express*, *USB*, *Ethernet*;
- багатоплатні кластерні прискорювачі, побудовані на базі багатьох ПЛІС, наприклад, *COPACOBANA (Cost Optimized Parallel COde Breaker)* компанії *SciEngines* – кластерний прискорювач, побудований компанією *SciEngines* на базі багатьох ПЛІС, який характеризується невисокою вартістю [9] і обладнаний 120 кристалами ПЛІС *Xilinx Spartan 3-1000*, *HS21 BladeCenter* [18] компанії *IBM*, або суперкомп'ютер *Maxwell* на основі програмовних логічних інтегральних схем, розроблений альянсом *FHPCA* [12]. Такі прискорювачі також взаємодіють з універсальним комп'ютером через периферійний інтерфейс.

3. Організація функціонування РККС з тіснозв'язаною реконфігуровною логікою

Архітектура РККС на основі універсальних процесорів з тіснозв'язаними прискорювачами передбачає пряме під'єднання прискорювачів до процесорів. Яскравими представниками прискорювачів з тіснозв'язаною архітектурою є співпроцесори, які встановлюються в роз'єм процесора. Платформи комп'ютерів, які використовують зазначені процесори, характеризуються досить широкими можливостями інтеграції різних пристроїв. Після появи платформ з декількома процесорами стало можливим створення нових продуктів, що можуть бути встановлені в роз'єм процесора. На рис. 2 наведено схему підключення основних елементів платформи *AMD*, на якій зображено різні варіанти підключення прискорювачів на базі ПЛІС [5].

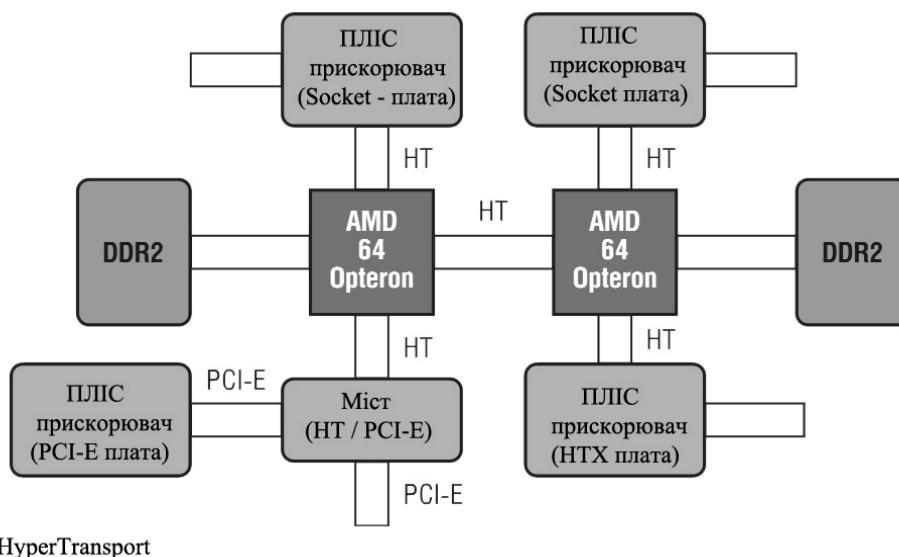


Рис. 2. Схема підключення прискорювачів до шини HyperTransport (платформа AMD)

Для даної платформи можливі наступні варіанти підключень:

- Плата з інтерфейсом *HT* (*HyperTransport* – двонаправлена послідовно/паралельна комп'ютерна шина з високою пропускнуою спроможністю і малими затримками) [15], яка безпосередньо встановлюється в процесорний роз'єм (на рисунку 2.7 позначена як *Socket-плата*).
- Плата з інтерфейсом *HT*, яка встановлюється в роз'єм *HTX* (*HyperTransport eXpansion* – специфікація слотного типу інтерфейсу, який має пряме з'єднання з мікропроцесором через *HyperTransport*).
- Плата з інтерфейсом *PCI Express*, яка під'єднується до роз'єму *HT* через міст *HT/PCI-Express*.

Прикладна програма взаємодіє з прискорювачем з допомогою ядра *HyperTransport*, що є сукупністю апаратних та програмних засобів для управління шиною та підключеними до неї пристроями. На програмному рівні ядро *HyperTransport* представлено драйвером, через який прикладна програма може керувати прискорювачем, а також здійснювати з ним обмін інформацією. У свою чергу доступ до даних прискорювач здійснює за допомогою контролера

прямого доступу до пам'яті.

На рис. 3 наведено схему для підключення основних елементів платформи *Intel* [5].

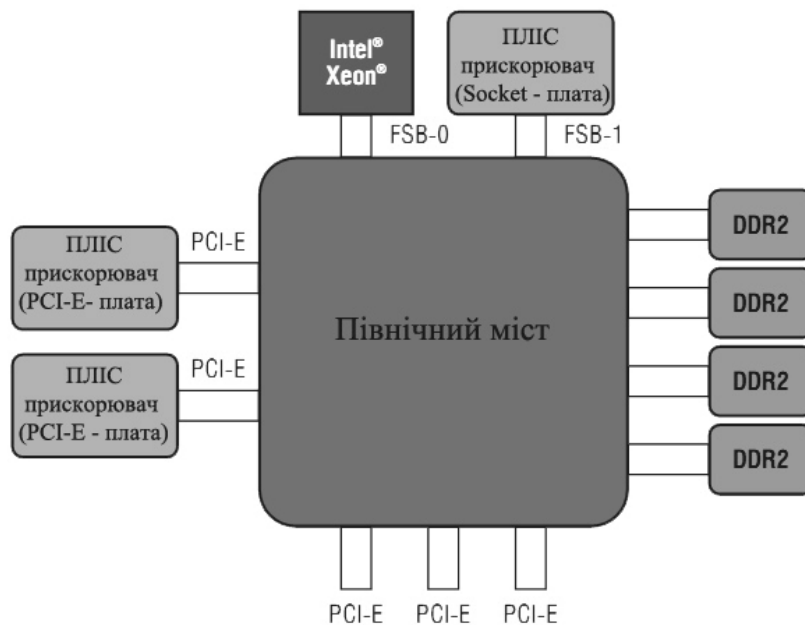


Рис. 3. Схема підключення прискорювачів у платформі Intel

Для даної платформи можливі наступні варіанти підключень:

- плата з інтерфейсом *FSB*, яка встановлюється безпосередньо в роз'єм процесора;
- плата з інтерфейсом *PCI Express*.

Прискорювачі з інтерфейсом *FSB* [13] встановлюються в роз'єм процесора платформи *Intel* [17]. *FSB* забезпечує високу швидкість та мінімальні затримки в порівнянні з іншими портами та шинами комп'ютера.

Є можливість підключення до комп'ютера не одного прискорювача, а декількох. В цьому випадку декілька модулів підключають між собою за кільцевою топологією через зовнішній, встановлений на платі, інтерфейс. Така схема підключення називається стековою, коли один модуль встановлюється поверх іншого, і забезпечує високий ступінь паралелізації, що є одним з ключових моментів у нарощенні продуктивності обчислювальної системи.

В архітектурі РККС на основі універсальних процесорів з тіснозв'язаними прискорювачами використовуються:

- тісноінтегровані в мережну структуру прискорювачі, зокрема реконфігуровні процесорні блоки *RP100* фірми *DRC* [11], реконфігуровний прискорювач *RCHTX* фірми *Celoxica* [7]. Тут передбачається пряме під'єднання прискорювачів до процесорів;
- тісноінтегровані в мережну структуру прискорювачі з прямим з'єднанням з мережними внутрішніми зв'язками. Ця архітектура також передбачає використання двох окремих мереж: процесорної та ПЛІС, але з їх накладанням, наприклад, з під'єднанням ПЛІС безпосередньо до мережного інтерфейсного кристала (*network interface chip – NIC*). Різниця тут полягає в тому, що ПЛІС під'єднані безпосередньо до основної мережі через *NIC*'и, що дає можливість одному процесору мати доступ до однієї або більше ПЛІС, тобто будь-яка ПЛІС може зв'язатися безпосередньо з будь-яким процесором. Прикладом такої архітектури є система *XDI* фірми *Cray* [22];
- прискорювачі з прямим з'єднанням з пристроями пам'яті. Тут прискорювачі з'єднані безпосередньо з системною шиною розподіленої пам'яті. Цим самим створюється мережа з топологією типу „кожен з кожним” і вимагається проведення диспетчеризації апаратних засобів для підтримки когерентності з'єднаних блоків пам'яті та пристроїв. Прикладом такої архітектури є архітектура *RASC (Reconfigurable Application Specific Computing)* [21] фірми *SGI* та її комунікаційна структура *NUMALink* [20];
- прискорювачі на основі архітектури *IMPLICIT+EXPLICIT* [16] фірми *SRC*, яка є гібридом вище згаданих типів архітектури.

Останні два типи архітектури передбачають з'єднання прискорювачів безпосередньо з системною шиною розподіленої пам'яті. Цим самим створюється мережа з топологією типу

“кожен з кожним” і вимагається високий рівень диспетчеризації апаратних засобів для підтримки когерентності з'єднаних блоків пам'яті та прискорювачів. Головним недоліком цього підходу є висока вартість.

4. Експериментальні дослідження характеристик реконфігурованих комп'ютерних систем залежно від складності задачі

Як показали висвітлені в [1, 2, 3] експериментальні дослідження слабозв'язаних та тіснозв'язаних реконфігурованих прискорювачів, організація зв'язку між ними і комп'ютером значній мірі визначає доцільного їх застосування в тій чи іншій області. Дослідження проводились на наступних платформах:

- процесор *Intel Celeron 2.0* ГГц; ОЗП 1 Гб; ОС *Windows XP Professional SP2*;
- процесор *AMD Athlon 64XP 2.0* ГГц; ОЗП 1 Гб; ОС *Windows XP Professional SP2*.

Для оцінювання слабозв'язаної архітектури використано прискорювач *XPRESSFX100-11* від компанії *PLDA* на ПЛІС *Virtex4 FX100* компанії *Xilinx*. Для оцінювання тіснозв'язаної архітектури використано таку саму ПЛІС *Virtex4 FX100* компанії *Xilinx* з розрахунком її підключення до високошвидкісної шини пропускною здатністю 16 Гб/с, яка при досліджуваних об'ємах даних не створює затримки передачі. В ПЛІС прискорювача реалізовано контролер інтерфейсу *PCI-Express* та спеціалізований процесор виконання тестового алгоритму.

Оцінювання було виконано на алгоритмах:

- перетворення *RGB* → *YUV*;
- цифрової фільтрації;
- дискретного косинусного перетворення (ДКП);
- швидкого перетворення Фур'є (ШПФ) на 1024 точки.

Апаратні витрати на реалізацію спеціалізованих процесорів в ПЛІС прискорювача склали:

- перетворення *RGB* → *YUV* – 22% площі ПЛІС;
- цифровий фільтр – 23% площі ПЛІС;
- дискретне косинусне перетворення – 29% площі ПЛІС;
- швидке перетворення Фур'є – 35% площі ПЛІС,

з яких 17% припадає на реалізацію контролера *PCI-Express*.

Передавання даних на прискорювач *PLDA XPRESSFX100-11* відбувається по 8 байт. Реалізовані в прискорювачі спеціалізовані процесори мають таку кількість вхідних та вихідних портів:

- перетворення *RGB* → *YUV* – 3 вхідних та 3 вихідних порти;
- цифровий фільтр – 4 вхідних, 4 вихідних;
- ДКП – 8 вхідних, 8 вихідних;
- ШПФ-1024 – 2 вхідних, 2 вихідних.

Обмін даними з прискорювачем було організовано декількома каналами. На графіках нижче (рис. 4 і 5) наведено залежності часу виконання зазначених алгоритмів на слабозв'язаному (рис. 4) та тіснозв'язаному (рис. 5) прискорювачах від обсягу вхідних даних при їх передачі одним каналом та декількома.

З наведених графіків видно, що частка тривалості пересилання даних є визначальною при значних об'ємах цих даних, причому при виконанні деяких алгоритмів на слабозв'язаному прискорювачі коефіцієнт прискорення $\epsilon < 1$ (тобто спостерігається програв в продуктивності порівняно з універсальним процесором). Це стосується перетворення *RGB* → *YUV*, причому у випадку передачі даних одним каналом та трьома, та алгоритму цифрової фільтрації у випадку передачі даних одним каналом. Для алгоритму ШПФ-1024 коефіцієнт прискорення становить 250 разів для тіснозв'язаної архітектури проти 10 разів для слабозв'язаної архітектури, а для алгоритму ДКП – 120 проти 2,5.

Виходячи з наведених вище результатів експериментальних досліджень виглядає доцільним перед вибором алгоритмів, які мають бути реалізовані в прискорювачі, виконати попереднє оцінювання ефективності такого рішення. Розглянемо далі як це можна зробити.

5. Методика визначення доцільності застосування прискорювача в реконфігурованій комп'ютерній системі

Під час визначення доцільності застосування прискорювача для виконання деякої програми P_{in} , яка передбачає опрацювання даних об'ємом Q_{data} , необхідно дослідити виконання наступного

співвідношення:

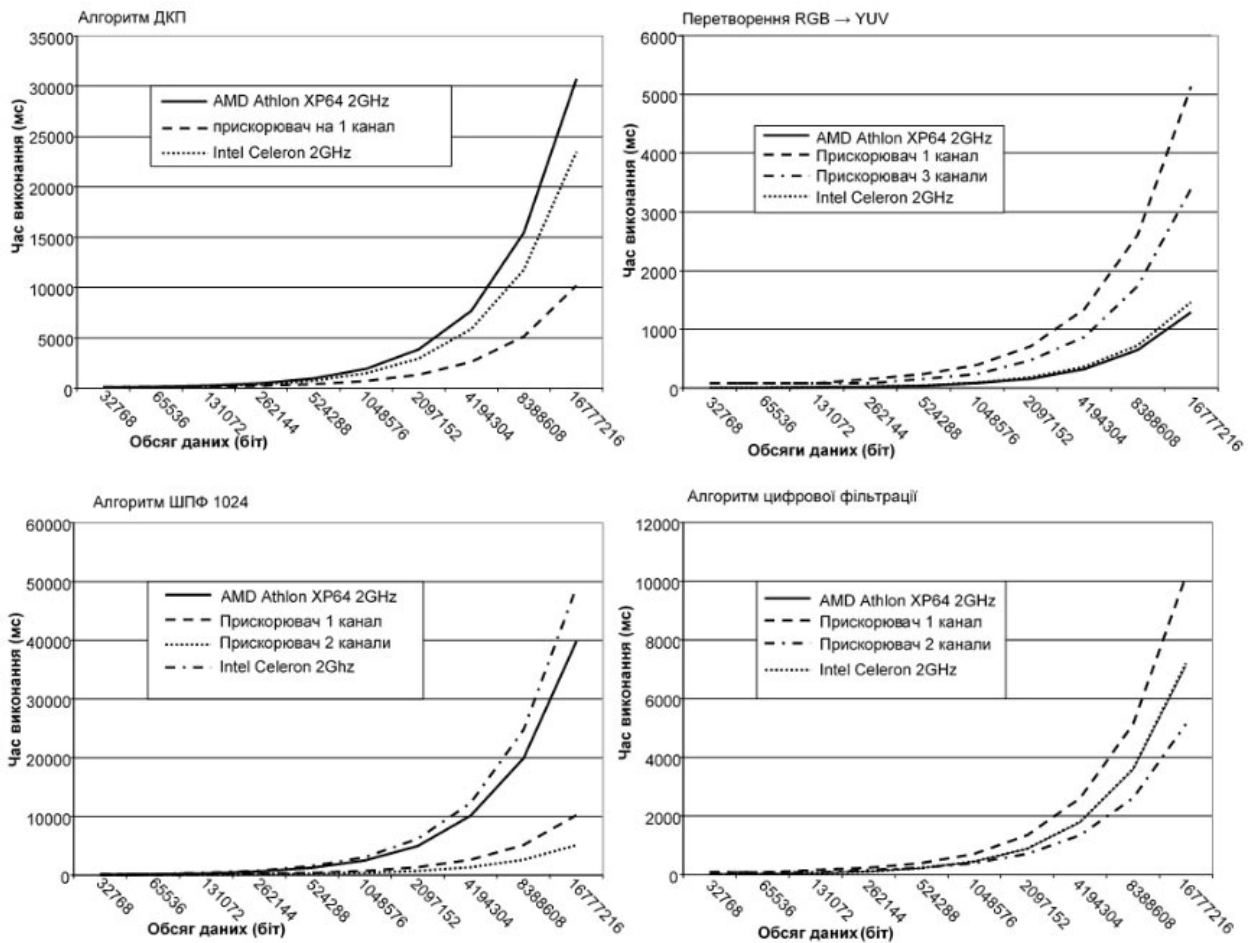


Рис. 4. Графіки залежності часу виконання алгоритмів на слабозв'язаному прискорювачі від обсягу вхідних даних при їх передачі одним каналом та декількома

$$\frac{Q_{data}}{V_{int}} < T_{UPP}(P_{in}) - T_{UPP}(P_{UPP}) + T_{SP}(P_{RE}), \quad (2)$$

де Q_{data} – кількість даних, що пересилаються між універсальним комп'ютером та прискорювачем; V_{int} – пропускна спроможність інтерфейсу, через який ці дані пересилаються; $T_{UPP}(P_{in})$, $T_{UPP}(P_{UPP})$ – тривалість виконання, відповідно, програми P_{in} та підпрограми P_{UPP} універсальним процесором UPP ; $T_{SP}(P_{RE})$ – тривалість виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP , синтезованим в реконфігурованому середовищі РККС.

Тривалість виконання програм P_{in} та P_{UPP} на універсальному комп'ютері можна визначити, знаючи характеристики обчислювальної платформи, з наступного виразу:

$$T_{UPP}(x) = \sum_{i=1}^{IQ(x)} \left(\frac{qclk(I_i)}{F_{UPP}} \right), \quad x \in (P_{in}, P_{UPP}), \quad (3)$$

де: $qclk(I_i)$ – кількість тактів виконання i -ї інструкції тієї вітки програми, яка має найвищу програмну складність; IQ – кількість інструкцій у вітці програми з найвищою програмною складністю; F_{UPP} – тактова частота роботи процесора універсального комп'ютера.

Тривалість $T_{SP}(P_{RE})$ виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP можна визначити на основі отриманих від засобів логічного синтезу даних про максимальну тактову частоту F_{SP} роботи цього процесора, та знаючи розрядність DW_{SP} слів даних, які він обробляє і кількість $qclk(SP)$ тактів, необхідних для їх обробки (ці дані

визначаються на етапі створення програмної моделі процесора):

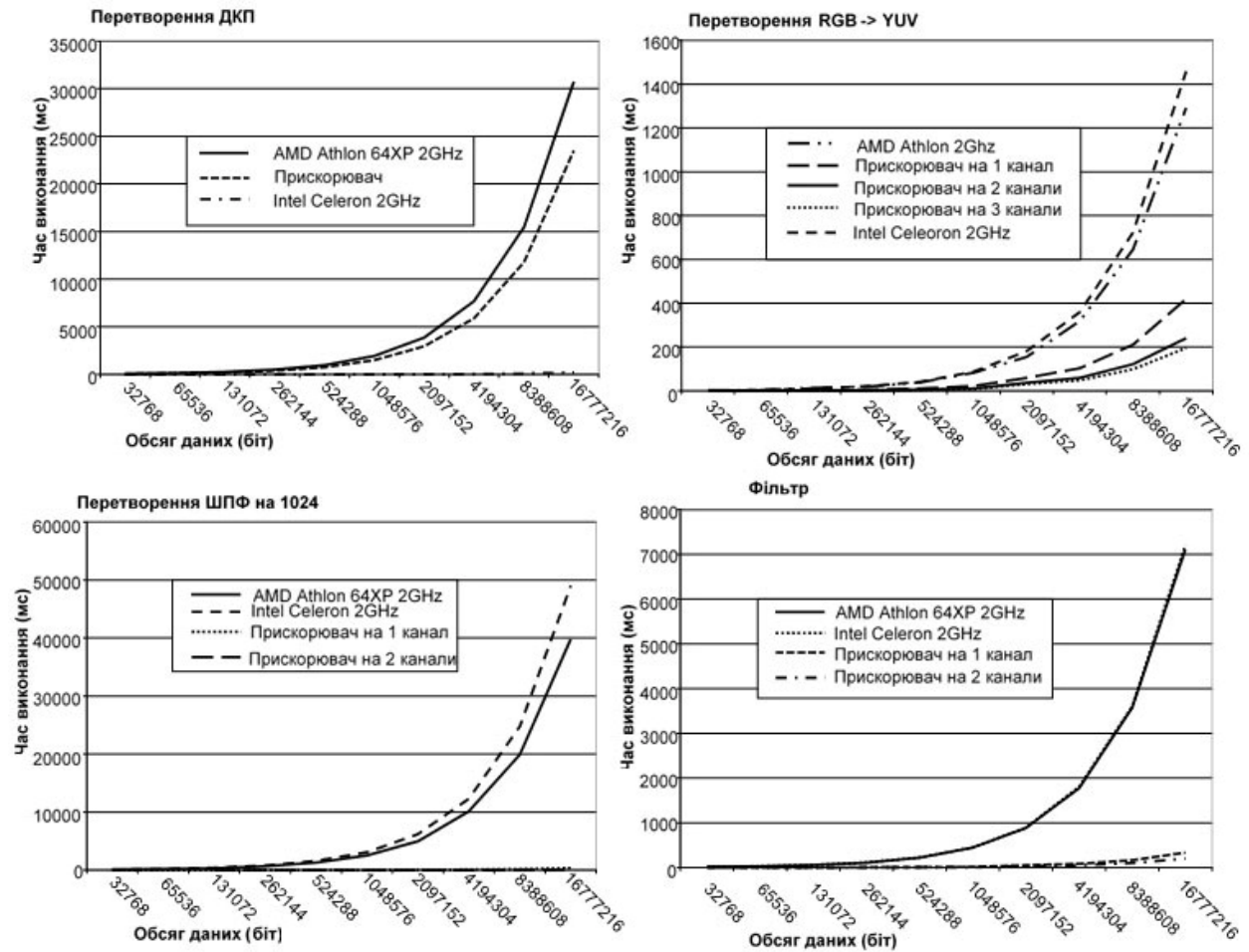


Рис. 5. Графіки залежності часу виконання алгоритмів на тіснозв'язаному прискорювачі від обсягу вхідних даних при їх передачі одним каналом та декількома

$$T_{SP}(P_{RE}) = \frac{qclk(SP)}{F_{SP} \cdot DW_{SP}} \quad (4)$$

Підставивши дані у вирази (3) і (4), можна обчислити коефіцієнт ефективності K_E застосування прискорювача для виконання програми P_{in} з позиції організації зв'язку між ним і комп'ютером:

$$K_E = \frac{Q_{data}}{V_{int} \cdot \left(\sum_{i=1}^{IQ(P_{in})} \left(\frac{qclk(I_i)}{F_{UPP}} \right) - \sum_{i=1}^{IQ(P_{UPP})} \left(\frac{qclk(I_i)}{F_{UPP}} \right) + \frac{qclk(SP)}{F_{SP} \cdot DW_{SP}} \right)}, \quad (5)$$

Таким чином, відповідно до запропонованої методики, визначення доцільності застосування прискорювача передбачає виконання наступних кроків:

1. визначення тривалості $T_{UPP}(P_{in})$ та $T_{UPP}(P_{UPP})$ виконання програм P_{in} та P_{UPP} на універсальному комп'ютері за виразом (3);
2. визначення тривалості $T_{SP}(P_{RE})$ виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP за виразом (4);
3. обчислення коефіцієнта ефективності K_E застосування прискорювача для виконання програми P_{in} за виразом (5).

Значення коефіцієнта ефективності повинно знаходитись в межах $0 < K_E < 1$, причому чим меншим буде це значення, тим вищою буде ефективність застосування прискорювача для виконання програми P_{in} .

Висновки

Проведено дослідження та порівняльний аналіз типів архітектури РККС. Показано, що для нарощення ресурсів та підвищення продуктивності, а також надання РККС властивості масштабованості, РККС будують шляхом під'єднання до процесора зовнішніх по відношенню до нього пристроїв реконфігуровної логіки, що є реалізацією тіснозв'язаної та слабозв'язаної типів їх архітектури. і обґрунтовує застосування реконфігуровних прискорювачів для побудови РККС.

Сформульовано основи організації функціонування РККС з слабозв'язаною реконфігуровною логікою та проведено оцінювання технічних характеристик існуючих реконфігуровних прискорювачів цих РККС.

Сформульовано основи організації функціонування РККС з тіснозв'язаною реконфігуровною логікою, проведено оцінювання технічних характеристик існуючих реконфігуровних прискорювачів цих РККС, а саме: тісноінтегрованих в мережну структуру прискорювачів, тісноінтегрованих прискорювачів з прямим під'єднанням до мережного інтерфейсного кристалу, прискорювачів з прямим з'єднанням з пристроями пам'яті та гібридних прискорювачів.

З метою визначення областей доцільного застосування слабозв'язаних та тіснозв'язаних реконфігуровних прискорювачів з позиції організації зв'язку між ними і комп'ютером проведено експериментальні дослідження характеристик РККС залежно від складності задачі та встановлено,

що при виконанні тестових алгоритмів з обчислювальною складністю $O\left(\frac{N}{2} \cdot \log N\right)$ на базі слабозв'язаної архітектури досягається прискорення на один порядок, а на базі тіснозв'язаної – на два порядки. У випадку виконання обчислювально складніших алгоритмів показники прискорення вищі. Разом з тим дослідження показали, що алгоритми, які характеризуються невисокою обчислювальною складністю та інтенсивним потоком даних, на зразок перетворення $RGB \rightarrow YUV$, на базі слабозв'язаної архітектури виконувати недоцільно.

Показано методику визначення доцільності застосування прискорювача з позиції організації зв'язку між ним і комп'ютером.

1. Мельник А.О. Використання реконфігуровних прискорювачів для підвищення продуктивності персональних комп'ютерів / А.О. Мельник, В.А. Мельник, З.Т. Сарайрех // Науковий вісник Чернівецького ун-ту. Комп'ютерні системи та компоненти. – Чернівці: Чернівецький національний університет імені Юрія Федьковича, 2010. - Т.1 Вип.1. – С. 20-25.
2. Мельник А.О. Реалізація програмних спеціалізованих процесорів в реконфігуровних прискорювачах універсальних комп'ютерів / А.О. Мельник, В.А. Клименко, Л.О. Цигилик, А.В. Юрчук // Вісник НУ "Львівська політехніка" "Комп'ютерні системи та мережі", –2009. –№ 609. –С. 69-77
3. Мельник А.О. Хамелеон – система високорівневого синтезу спеціалізованих процесорів / А.О. Мельник, В.А. Клименко, Л.О. Цигилик, А.В. Юрчук // Радіоелектронні і комп'ютерні системи, Харків «ХАІ». –2009. –С. 189-193.
4. Мельник В.А. Основи побудови та організації функціонування реконфігуровних комп'ютерних систем / В.А. Мельник // Науковий журнал «Вісник Хмельницького Національного Університету», Серія «Технічні науки». – №6. – 2012 р. – С. 212 – 217.
5. Шматок А. Аппаратные ускорители приложений на базе ПЛИС / А. Шматок // Современная электроника, -№6, 2007. – С. 60-63.
6. AMBA Specification (Rev 2.0). ARM Limited, 1999. – 230 p.
7. Celoxica Ltd. RCHTX-XV4 High Performance Computing (HPC) Application Acceleration Board Datasheet. Version 1.0. 2006 [Електронний ресурс] // – Режим доступу: http://www.hypertransport.org/docs/tech/rchtx_datasheet_screen.pdf
8. ClearSpeed Technology. ClearSpeed Advance X620 Accelerator Card User Guide. // Document No. 60-UG-1302 Revision: 3.E September 2008. -16p. [Електронний ресурс] // – Режим доступу: http://support.clearspeed.com/resources/documentation/X620_Advance_Card_User_Guide_Rev3.E.pdf
9. СОРАСОВАНА – Special Purpose Hardware for Code Breaking [Електронний ресурс] // – Режим доступу: <http://www.copacobana.org>

10. Devlin Malachy. DIME – The First Module Standard for FPGA Based High Performance Computing / Malachy Devlin, Allan J. Cante // FPL '99 Proceedings of the 9th International Workshop on Field-Programmable Logic and Applications. Springer-Verlag, London, UK, 1999. – P. 425 – 430.
11. DRC Computer Corporation. RPU100-L60 DRC Reconfigurable Processor Unit. A breakthrough in coprocessor technology [Електронний ресурс] // – Режим доступу: http://www.drccomputer.com/pdfs/DRC_RPU100_datasheet.pdf
12. FHPCA Supercomputer: Maxwell [Електронний ресурс] // – Режим доступу: <http://www.epcc.ed.ac.uk/projects-portfolio/fhpca-supercomputer-maxwell>
13. FPGA Accelerated Computing Solutions. Intel Xeon Front Side Bus [Електронний ресурс] // – Режим доступу: http://www.nallatech.com/images/stories/product/facs/fsb-accelerator-module/FSB_Accelerators_v1-5.pdf
14. H100 Series FPGA Application Accelerators. Version 1.9. September 2008 // [Електронний ресурс] // – Режим доступу: <http://www.skyblue.de/nallatech/5595.pdf>
15. HyperTransport Consortium. "HyperTransport Link Specification", 2005.
16. IMPLICIT+EXPLICIT ARCHITECTURE [Електронний ресурс] // – Режим доступу: <http://www.srccomp.com/techpubs/implicitexplicit.asp>
17. Intel Xeon Processor E5 Family [Електронний ресурс] / – Режим доступу: http://www.intel.com/products/processor/xeon5000/index.htm?iid=servproc+body_xeon5000
18. International Business Machines Corporation. BladeCenter HS21 Type 8853 Installation and User's Guide, Fourth Edition (August 2007) - 66p. // [Електронний ресурс] // – Режим доступу: http://publib.boulder.ibm.com/infocenter/bladectr/documentation/topic/com.ibm.bladecenter.hs21.doc/hs21_install_ug.pdf
19. Scott Hauck, André DeHon. "Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation"/ Morgan Kaufmann, 2008. – 944 p.
20. SGI NUMALink Interconnect Fabric [Електронний ресурс] // – Режим доступу: <http://www.sgi.com/products/servers/numalink.html>
21. Silicon Graphics, Inc. Reconfigurable Application-Specific Computing User's Guide. 2004-2006, 196p. [Електронний ресурс] // – Режим доступу: <http://techpubs.sgi.com/library/manuals/4000/007-4718-004/pdf/007-4718-004.pdf>
22. Ulmer, Craig D., Hilles, Ryan, Thompson, David C. Reconfigurable computing aspects of the Cray XD1. Presentation at the Cray User Group (CUG 2005), May 16-19, 2005. Albuquerque, USA. -11p. [Електронний ресурс] // – Режим доступу: http://www.craigulmer.com/portfolio/unlocked/050516_cug_rc_aspects_of_xdl.pdf
23. Wade D. Peterson, Silicore Corporation. Specification for the WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores. Revision: B.3, Released: September 7, 2002. – 140 p.