

подземных трубопроводов // Прогрессивные материалы, технологические процессы и оборудование для защиты металлов от коррозии. К., 1990. С.85-87. 4. Патент 1804636 (SU). МКИ G01V3/11. Устройство для определения расположения магистральных трубопроводов / Р.М. Джала, Б.Я. Вербенец, А.А. Андреев, Л.М. Пеккер. Открытия. Изобрет. 1993. № 11. 5. Барон Д.А., Горднев И.И., Евдокимов В.Н. и др. Строительство кабельных сооружений связи. М., 1998. 6. Патент 10884А (UA). G01R19/00, G01R33/02, G01V3/10. Генератор і обстеження підземних комунікацій / Драбич П.П. Открытия. Изобрет. 1996. № 4. 7. Мизюк Л.Я. Входные преобразователи для измерения напряженности низкочастотных магнитных полей. К., 1964. 8. Драбич П.П. Поиск и обследование протяженных электропроводящих коммуникаций // Измерительная техника. М., 1996. № 3. С.56-58. 9. Улич Г.Г., Ревы Р.У. Коррозия и борьба с ней. Введение в коррозионную науку и технику. Л., 1989.

УДК 621.382

Мельник А.О., Ахмад Аль-Кхатіб
НУ “Львівська політехніка”, кафедра ЕОМ

АРХІТЕКТУРА ПРОЦЕСОРІВ ШВИДКИХ ОРТОГОНАЛЬНИХ ПЕРЕТВОРЕНЬ

© Мельник А.О., Ахмад Аль-Кхатіб, 2000

Застосовано програмно- та апаратно-орієнтовані процесори для ортргонального перетворення Фур'є. Описано особливості реалізації даного перетворення за допомогою процесорів з різними архітектурами.

Вступ

При моделюванні та розробці систем обробки даних різного призначення широко застосовують методи, що ґрунтуються на математичному апараті ортогональних перетворень (ОП) – Фур'є, Хартлі, Гільберта, Лапласа, Френеля, косинусного, синусного тощо. На їх основі будують процедури спектрально-кореляційного аналізу, фільтрації, кепстрального аналізу, кодування сигналів та зображень, а також ідентифікації лінійних та нелінійних систем, відтворення зображень за проєкціями, аналізу та синтезу цифрових голограм тощо. При цьому на виконання ОП припадає значна частина загальної кількості обчислень і необхідних для цього ресурсів, тому ефективність їх виконання суттєво впливає на покращення характеристик цілого ряду технічних систем.

Для скорочення об'ємів обчислень, необхідних для виконання алгоритмів ОП, використовуються алгоритми швидких ортогональних перетворень (ШОП), для реалізації яких створюються спеціалізовані процесори, в архітектурі яких враховані особливості цих алгоритмів. На сьогодні сформувалися два основні підходи до побудови процесорів ШОП [1]. *Перший* ґрунтується на використанні універсальних програмованих процесорів, а особливості виконуваних задач враховуються, по суті, шляхом спеціалізації їх програмного забезпечення. *Другий* підхід базується на використанні процесорів, орієнтованих на виконання алгоритмів ШОП апаратним способом. Такі процесори називають функціонально-або апаратно-орієнтованими (в англійській літературі – ASIC – *Application-Specific Integration Circuit*). Розглянемо архітектури та методи проектування процесорів ШОП на основі обох підходів більш детально.

1. Архітектура програмованих процесорів для реалізації ШОП

1.1. Реалізація ШОП на програмованих процесорах з повним та зі скороченим набором команд

Процесори з повним набором команд (ППНК, англійський термін *CISC – Complex Instruction Set Computer*) та скороченим набором команд (ПСНК – англійський термін *RISC – Redused Set Computer*) сьогодні становлять переважну частину програмованих процесорів, які випускаються промисловістю [2]. Для них напрацьована значна кількість програмного забезпечення та необхідних для проектування технологічних засобів. Разом з тим, орієнтація архітектури цих процесорів на використання в універсальних комп'ютерах вимагає введення в них спеціальних засобів для роботи в цих комп'ютерах, що робить їх надлишковими для побудови процесорів ШОП. Крім того, вони не завжди забезпечують потрібну продуктивність, оскільки в цих процесорах лише для виконання однієї базової операції довільного алгоритму ШОП необхідно застосувати десятки команд.

1.2. Реалізація ШОП на програмованих процесорах з широким форматом команди

Програмовані процесори з широким форматом команди (ПШФК, англійський термін *VLIW – Very Long Instruction Word*) передбачають паралельну роботу великої кількості функціональних блоків. Ці блоки можуть керуватись незалежно, що передбачає наявність в форматі команди великої кількості розрядів, згрупованих в ортогональні поля. Сьогодні промисловість випускає декілька типів ПШФК. Це, зокрема, один з найпродуктивніших процесорів TMS320C60 фірми *Texas Instruments* [3].

Архітектура ПШФК дозволяє скоротити об'єм обладнання, яке необхідне для паралельної видачі декількох команд, і потенційно чим більша кількість команд видається паралельно, тим більша ця економія. Наприклад, суперскалярна машина, яка забезпечує паралельну видачу двох команд, вимагає паралельного аналізу двох кодів операцій, шести полів номерів регістрів, а також того, щоб динамічно аналізувалась можливість видачі однієї або двох команд і виконувався розподіл цих команд по функціональних пристроях. Хоча вимоги до об'єму апаратури для паралельної видачі двох команд залишаються досить помірними і можна навіть збільшити ступінь розпаралелювання до чотирьох (що застосовується в сучасних мікропроцесорах), подальше збільшення кількості виданих паралельно для виконання команд призводить до зростання складності реалізації через необхідність визначення порядку чергування команд і існуючих між ними залежностей.

Архітектура ПШФК базується на багатьох незалежних операційних пристроях. Замість того, щоб паралельно видавати в ці пристрої незалежні команди, в таких процесорах декілька операцій вкладаються в одну дуже довгу команду. При цьому відповідальність за вибір паралельно виданих для виконання операцій повністю лягає на компілятор, а апаратні засоби, необхідні для реалізації суперскалярної обробки, просто відсутні. Команда ПШФК може містити, наприклад, дві цілочислові операції, дві операції з плаваючою крапкою, дві операції звернення до пам'яті і операцію переходу. Така команда буде мати набір полів для кожного операційного пристрою, можливо від 16 до 24 біт на пристрій, що збільшує довжину команди від 112 до 168 біт.

Для ПШФК був розроблений новий метод планування видачі команд, названий "траєкторіальним плануванням". При використанні цього методу із послідовності вихідної програми генеруються довгі команди шляхом перегляду програми за межами базових блоків. (Базовий блок - це лінійний відрізок програми без відгалужень).

З точки зору архітектурних ідей ПШФК можна розглядати як розширення ПЗСК. Як і в архітектурі ПЗСК, апаратні ресурси ПШФК надані компілятору і плануються статично. В ПШФК до цих ресурсів належать конвеєрні операційні пристрої, шини і банки пам'яті. Для підтримки високої пропускної здатності між операційними пристроями і регістрами необхідно використати декілька наборів регістрів. Апаратне вирішення конфліктів виключається, і перевага надається простій логіці керування. На відміну від традиційних процесорів регістри і шини не резервуються, а їх використання повністю визначається під час компіляції. В ПШФК, крім того, цей принцип заміни керування під час виконання програми на планування під час компіляції, поширений на системи пам'яті. Для підтримки зайнятості конвеєрних операційних пристроїв треба забезпечити високу пропускну здатність пам'яті. Одним із методів збільшення пропускної здатності пам'яті є її розшарування. Однак в системі з розшарованою пам'яттю виникає конфлікт банку, якщо банк зайнятий попереднім зверненням. В ПЗСК стан зайнятості банків пам'яті відслідковується апаратно і перевіряється, коли видається команда, виконання якої пов'язано зі зверненням до пам'яті. В ПШФК ця функція передана програмним засобам. Можливі конфлікти банків визначає спеціальний модуль компілятора - модуль запобігання конфліктам. Виявлення конфліктів не є задачею оптимізації, це скоріше функція контролю коректності виконання операцій. Компілятор повинен бути здатним визначити, що конфлікти неможливі або, у протилежному випадку, допускати, що може виникнути найгірша ситуація. У визначених ситуаціях, наприклад, у тому випадку, коли виконується звернення до масиву, а індекс обраховується під час виконання програми, немає простого рішення. Якщо компілятор не може визначити, що конфлікт не виникне, операції не можуть плануватися для паралельного виконання, а це веде до зниження продуктивності.

Компілятор із трасувальним плануванням визначає ділянку програми без зворотніх дуг (переходів назад), яка стає кандидатом для складання розкладу. Зворотні дуги звичайно наявні в програмах з циклами. Для збільшення розміру тіла циклу широко використовується методика розкрутки циклів, що приводить до утворення великих фрагментів програми, які не мають обернених дуг. Якщо програма має тільки переходи вперед, компілятор робить евристичне передбачення вибору умовних віток. Шлях, що має найбільшу імовірність виконання (його називають трасою), використовується для оптимізації, що проводиться з врахуванням залежностей за даними між командами і обмежень апаратних ресурсів. Під час планування генерується довге командне слово. Всі операції довгого командного слова видаються одночасно і виконуються паралельно. Після обробки першої траси планується наступний шлях, що має найбільшу імовірність виконання (попередня траса більше не розглядається). Процес упаковки команд послідовної програми в довгі командні слова продовжується доти, поки не буде оптимізована вся програма.

Ключовою умовою досягнення ефективної роботи ПШФК є коректне передбачення вибору умовних віток. Відзначено, наприклад, що прогноз умовних віток для наукових програм часто виявляється точним. Повернення назад присутні в усіх ітераціях циклу, за виключенням останньої. Отже, "прогноз", який уже надається самими переходами назад, буде коректним в більшості випадків. Інші умовні вітки, наприклад вітка обробки переповнення і перевірки граничних умов (вихід за межі масиву), також надійно передбачувані.

Алгоритми ШОП виконуються на ПШФК досить швидко і ефективно, оскільки хід обчислювального процесу в них не залежить від даних і може бути спланований наперед. Разом з тим, вони вимагають деяких додаткових затрат часу на організацію обчислюваль-

ного процесу. Крім того, ПШФК мають ітераційну структуру і також в багатьох випадках не задовольняють вимоги за продуктивністю. Так, в операційному пристрої процесора TMS320C60, який має 8 паралельно включених арифметико-логічних пристроїв, для виконання базової операції алгоритму ШПФ комплексних чисел необхідно виконати декілька широкоформатних команд.

1.3. Реалізація ШОП на програмованих процесорах обробки сигналів

Програмовані процесори обробки сигналів (ППОС, англійський термін *PDSP – programmable digital signal processors*) мають архітектуру, орієнтовану на вирішення задач ЦОС, включаючи ШОП. При побудові ППОС може бути використана будь-яка з описаних раніше концепцій (ПРСК, ПЗСК, ПНФК). ППОС, зберігаючи універсальність, мають вищі параметри за продуктивністю завдяки вузлам, орієнтованим на виконання задач ЦОС. Звичайно процесор належить до ППОС, коли він має такі основні риси [4]:

- розділені шини даних і команд із забезпеченням інформаційного обміну між ними;
- широке використання конвеєрного принципу обробки даних;
- структурна орієнтація операційного пристрою на виконання операції накопичення добутків чисел. Він містить паралельний перемножувач та акумулятор в додаток до стандартного арифметико-логічного пристрою. Це дозволяє виконувати команду множення-накопичення за один машинний цикл;
- використання широкорозрядних блоків оперативної і постійної пам'яті великого об'єму з можливістю її секціювання із забезпеченням незалежного доступу до секцій;
- структурна орієнтація формувачів адрес даних на роботу з масивами даних та на виконувани алгоритми адресації пам'яті;
- включення в пристрої вибірки команд стеків індексних регістрів, регістрів загального призначення, регістрів попередньої вибірки та інших апаратних засобів для маніпуляцій з командами і даними;
- включення спеціальних команд для виконання алгоритмів ЦОС в систему команд ППОС;
- використання швидкодіючих послідовних і паралельних інтерфейсів, які забезпечують створення багатопроекторних систем;
- короткий командний цикл.

Завдяки цим рисам ППОС виконують алгоритми ШОП досить швидко. Особливо цьому сприяє наявність паралельного перемножувача та акумулятора, а також спеціальних формувачів адрес даних. Але ППОС мають ітераційну структуру і також в багатьох випадках не задовольняють вимоги за продуктивністю. Для виконання базової операції алгоритму ШПФ комплексних чисел необхідно виконати 10 команд множення-накопичення та більше десятка команд пересилання даних. Тому в останніх ППОС прослідковується тенденція до ускладнення тракту даних через введення паралельних операційних пристроїв.

1.4. Програмовані процесори із спеціалізованою системою команд для реалізації ШОП

Програмовані процесори із спеціалізованою системою команд (ПССК, англійський термін *ASIP – application-specific instruction-set processor*) можуть застосовуватися до виконання широкого класу алгоритмів в конкретній області, на яку вони орієнтовані. Разом з універсальністю архітектури її орієнтація на область застосування дозволяє оптимізувати

тракти даних, організацію пам'яті, а часто і інші функціональні вузли процесора. У результаті ПССК зазвичай містять кілька процесорних елементів або виконавчих блоків, що забезпечує більш високу продуктивність порівняно з ППОС. Хорошим прикладом ПССК може бути процесор відеосигналів VSP фірми NEC Corporation [5].

Залежно від виконуваного алгоритму ШОП до тракту даних ПССК доцільно ввести один або декілька операційних пристроїв для виконання відповідної базової операції. Для виконання базової операції за основою два алгоритму швидкого перетворення Фур'є (ШПФ) тракт даних ПССК повинен містити 4 паралельних перемножувачі та 6 суматорів. В ПССК для виконання алгоритму швидкого перетворення Хартлі (ШПХ) достатньо 4 паралельних перемножувачів та 2 суматори, а ПССК для виконання алгоритму швидкого косинусного перетворення (ШКП) достатньо 1 паралельного перемножувача та 2 суматорів [6].

Але ПССК, як і ППОС, мають ітераційну структуру і також в багатьох випадках не задовольняють вимоги за продуктивністю. Для виконання алгоритму ШОП у такому процесорі навіть при можливості виконання в його тракті обробки даних однієї базової операції необхідно виконати $N/2\log N$ базових операцій, кожна з яких містить крім арифметичних також команди формування адрес даних та команд, виборки команд, виборки, пересилки та зберігання даних. З метою прискорення обчислень в ПССК вводяться апаратно-орієнтовані процесори, архітектура яких оптимізована до виконання конкретних алгоритмів ШОП, наприклад ШКП, на базі яких виконуються інші ШОП [6].

1.5. Реалізація ШОП на паралельних системах на основі програмованих процесорів

На основі програмованих процесорів можуть бути реалізовані паралельні багато-процесорні системи різних архітектур. На сьогодні вже з'явилися перші варіанти однокристалльної реалізації таких систем.

Матричні паралельні процесори (МПП, англійський термін MPP – *matrix parallel processor*) містять велику кількість паралельно включених програмованих процесорів, з'єднаних комутаційною мережею конкретної архітектури, наприклад типу гіперкуб, типу багатоступінчастої мережі тощо. Як приклад МПП можна навести однокристалні багато-процесорні системи PADDI та PADDI 2 університету Берклі [7,8]. Зокрема МПП PADDI 2 має 48 процесорних елементів, в операційному пристрої кожного з яких враховані особливості задач ЦОС.

Процесори із спільною пам'яттю (ПСП, англійський термін SMP – *shared memory processor*) також складаються із великої кількості паралельно включених програмованих процесорів, які з'єднані між собою через спільну пам'ять. З деяким наближенням до такої системи можна віднести мультимедіа-відео-процесор TMS320C80 [9], в якому 4 паралельних ППОС взаємодіють між собою через спільну багатопортову пам'ять під управлінням одного ПСНК.

Однорідні обчислювальні середовища (ООС) подібно до МПП складаються із великої кількості паралельно включених однотипних універсальних елементарних процесорів, з'єднаних комутаційною мережею регулярної архітектури. Як приклад однокристалльної реалізації ООС можна навести керований даними масив процесорів, розроблений в Масачусетському університеті Ахмерста. Він передбачає розміщення на кристалі біля 100 гексагонально з'єднаних процесорних елементів. Подібна до структури графа ШОП архітектура комутаційної мережі описаного ООС сприяє ефективній реалізації на ньому перетворень такого типу.

2. Архітектура апаратно-орієнтованих процесорів для реалізації ШОП

2.1. Алгоритмічні процесори ШОП

Апаратно-орієнтовані алгоритмічні процесори (АП) ШОП мають ітераційну архітектуру, орієнтовану на виконання конкретного алгоритму [10]. До складу АП входять такі вузли: операційний пристрій, багатопортова пам'ять та пристрій керування. Заданий алгоритм у такому процесорі виконується пропусканням за потрібну кількість разів даних через операційний пристрій, в якому виконуються базові операції алгоритму. Операційний пристрій може бути конвеєрним, що дозволяє суміщати в часі обробку різних даних. У більшості випадків управління АП є апаратним, що дозволяє досягти вищої продуктивності. При наявності в операційному пристрої АП k паралельно включених пристроїв для виконання базових операцій N -точкового алгоритму ШОП необхідно виконати $N(\log_p N)/pk$ тактів, де p – основа базової операції.

На сьогодні більшість однокристальних процесорів ШОП належать до класу АП. Наприклад, в найбільш продуктивних процесорах ШПФ комплексних чисел DSP-24 [11], BDSP9124 [12] та SPIFFEE [13] обчислення проводяться на операційному пристрої, який виконує базову операцію алгоритму ШПФ комплексних чисел. Для спрощення формування адрес даних в більшості процесорів цього типу використовуються графи ШОП з постійною структурою.

2.2. Граф-алгоритмічні процесори ШОП

В апаратно-орієнтованому граф-алгоритмічному процесорі (ГАП) алгоритм ШОП виконується одноразовим проходженням даних через його операційний пристрій [10]. В операційному пристрої апаратно відображається весь граф виконуваного алгоритму відповідним з'єднанням комбінаційних схем, кожна з яких виконує відповідний функціональний оператор алгоритму. У конвеєрному ГАП (КГАП) операційний пристрій є конвеєризованим, тобто комбінаційні схеми, що виконують операції кожного ярусу алгоритму, розділені регістровою пам'яттю. Крім операційного пристрою тут присутня вхідна буферна пам'ять, необхідна для впорядкування вхідних даних, та пристрій управління, який формує адреси даних та синхронізує роботу процесора.

На сучасному рівні розвитку інтегральної технології можлива реалізація в одному кристалі ГАП тільки для ШОП невеликих розмірів. Прикладом може бути процесор двовимірного 8-точкового косинусного перетворення [14].

2.3. Паралельні багатопроцесорні системи ШОП

На основі апаратно-орієнтованих процесорів можуть бути реалізовані паралельні багатопроцесорні системи різних архітектур. На сьогодні вже з'явилися перші варіанти однокристальної реалізації таких систем.

Апаратно-орієнтовані матричні паралельні процесори (МПП) містять велику кількість паралельно та (або) послідовно включених АП та ГАП, з'єднаних комутаційною мережею конкретної архітектури, яка визначається структурою виконуваного алгоритму. Тут алгоритм ШОП ділиться на паралельні незалежні вітки або на незалежні в часі етапи. Порядок з'єднання та кількість процесорів у кожному ярусі різні і залежить від алгоритму, для виконання якого будується система. Найбільш ефективними для виконання алгоритмів ШОП у реальному масштабі часу є МПП з послідовно з'єднаними ГАП, кількість яких дорівнює кількості етапів виконуваного алгоритму. Тут кожний ГАП виконує базову опера-

цію конкретного етапу алгоритму ШОП та здійснює необхідну перестановку даних для наступного етапу. У таких МПП необхідна тактова частота прийому та обробки даних досягається створенням КГАП з потрібною кількістю вхідних каналів [15].

Апаратно-орієнтовані процесори із спільною пам'яттю (ПСП) також містять велику кількість паралельно включених АП та ГАП, які з'єднані між собою через спільну пам'ять. За рахунок використання одного банку пам'яті тут вдається зменшити загальний її об'єм та прискорити виконання алгоритмів ШОП, оскільки скорочується кількість пересилок даних.

Для підвищення регулярності та однорідності структури АОП ШОП в останні роки багато розробок мали на меті їх реалізацію у вигляді систолічних структур (СС). СС подібно до МПП містять велику кількість паралельно включених однотипних спеціалізованих елементарних процесорів, з'єднаних комутаційною мережею регулярної архітектури. Як процесорний елемент систолічної структури можуть використовуватися однотипні ГАП або комбінаційна схема, яка може виконати довільний функціональний оператор алгоритму в операційному пристрої ГАП. У першому випадку ускладнюється розробка уніфікованого ГАП, а в другому випадку необхідно розробляти систолічний алгоритм, після чого задача зводиться до синтезу ГАП [16]. Створення АОП у вигляді систолічної структури спрощує процес проектування, хоча вимагає додаткового обладнання та може дещо зменшити продуктивність.

3. Програмована чи орієнтована на виконуваний алгоритм архітектура процесора ШОП – альтернатива вибору

Основна перевага програмованих процесорів універсальної архітектури - гнучкість. Запис програм виконання заданого набору алгоритмів в пам'ять програм дозволяє створити процесор ШОП із заданими параметрами. Такі процесори піддаються перепрограмуванню заміною вмісту пам'яті програм. При створенні відповідних трансляторів програмне забезпечення таких процесорів може бути написане мовами програмування високого рівня, що робить їх доступними для широкого кола користувачів. Крім того, дуже суттєвою перевагою цього підходу є можливість використання створених раніше програмних засобів. Разом з тим існують причини, через які використання програмованих процесорів універсальної архітектури для побудови процесорів ШОП може бути недоцільним.

По-перше – це висока трудомісткість розробки, оскільки до складу процесора ШОП крім самого програмованого процесора необхідно включити засоби для реалізації інтерфейсних функцій, синхронізації, розширення пам'яті програм і пам'яті даних тощо. Сам процес розробки вимагає створення необхідного програмного забезпечення, наявності або створення технологічних програмних засобів для відпрацювання програм, а також програмно-апаратних засобів для відлагодження апаратної та програмної частин процесора ШОП.

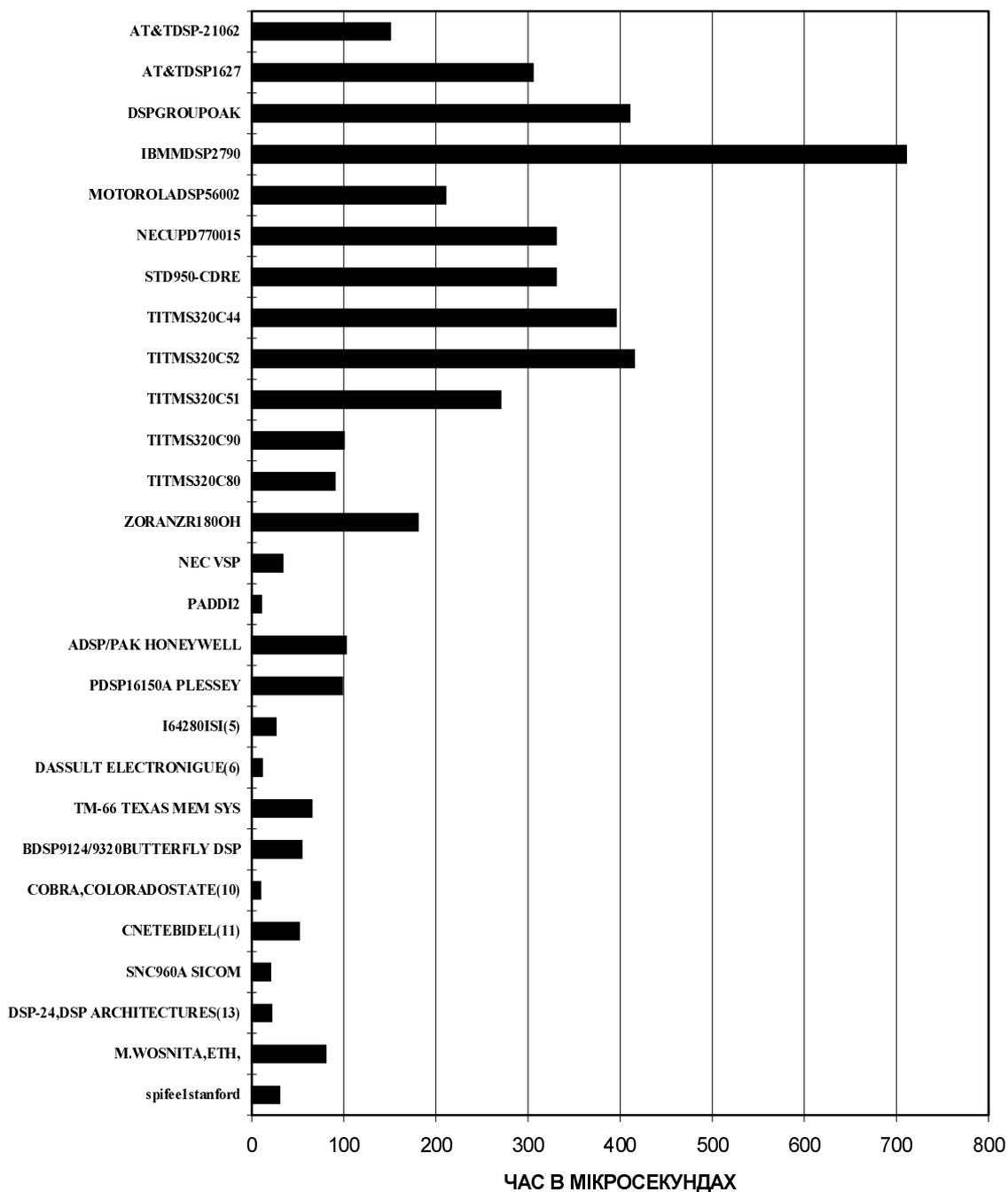
По-друге, універсальна архітектура є занадто надлишковою для виконання алгоритмів ШОП у функціональному і структурному відношеннях. Це призводить до надлишкової споживаної потужності, збільшення ступеня інтеграції, кількості виводів корпусів та розмірів кристалів.

По-третє, універсальний процесор може не задовольнити вимоги за продуктивністю. Забезпечити необхідну продуктивність можна шляхом побудови багатопроцесорних систем, але їх використання може бути занадто дорогим. Тут мається на увазі вартість не стільки власне багатопроцесорної системи, скільки вартість допоміжних засобів, що забезпечують використання такої системи.

Зараз, коли досягнення мікроелектронної технології підтримуються потужними САПР, другий підхід, що передбачає створення апаратно-орієнтованих на виконувани алгоритми процесорів, є реальною альтернативою універсальним програмованим процесорам ШОП.

По-перше, такий підхід забезпечує максимально можливу продуктивність при вирішенні заданої задачі на кристалі заданого об'єму.

По-друге, він вимагає мінімальних витрат обладнання на побудову процесора ШОП за рахунок відшукування компромісу між програмними та апаратними засобами.



Час виконання 1024-точкового ШПФ на програмованих та апаратно-орієнтованих процесорах різних архітектур

По-третє, процес проектування АОП є простішим порівняно з проектуванням програмованих універсальних процесорів, включаючи процес тестування.

Більше того, сучасні мови опису апаратних засобів, зокрема VHDL [17], мають настільки високий рівень, що процес проектування апаратно-орієнтованого процесора не є набагато складнішим, порівнюючи з розробкою спеціального програмного забезпечення для універсального процесора. Якщо врахувати ще можливості сучасних САПР щодо забезпечення повного відпрацювання моделей електронних компонент, включаючи їх роботу у складі процесора ШОП, а також використання бібліотек раніше створених електронних компонент, то стає очевидним, що настав час, коли другий підхід починає тіснити перший.

На рисунку наведений час виконання алгоритму 1024-точкового ШПФ комплексних чисел на програмованих та апаратно-орієнтованих процесорах описаних вище архітектур.

На основі проведеного аналізу архітектур та даних рисунку розглянуті архітектури процесорів можна розмістити в такій послідовності в порядку зростання їх потенційної продуктивності:

ППНК-ПСНК-ППОС-ПШФК-ПССК-АП-МПП-ПСП-СС-ГАП-КГАП.

4. Висновки

Проведено порівняльний аналіз використання програмованих та апаратно-орієнтованих процесорів для виконання ШОП. Розглянуті особливості виконання алгоритмів ШОП на програмованих процесорах різних архітектур: ПРСК, ПЗСК, ПШФК, ПССК, а також на паралельних системах на основі таких процесорів: МПП, ПСП, ООС. Розглянуті особливості побудови АОП для виконання алгоритмів ШОП різних архітектур: АП, ГАП, КГАП, а також на паралельних системах на основі таких процесорів: МПП, ПСП, СС. Оцінено переваги та недоліки використання процесорів різних архітектур, а також продуктивність програмованих та апаратно-орієнтованих процесорів при виконанні ШОП.

1. Мельник А.О., Тарасенко В.П.. *Сучасні ситуативно-методологічні аспекти створення спеціалізованих комп'ютерних систем. Наукові вісті. Національний технічний університет КПІ. 1997. № 1. С.18-21.*
2. Patterson D., Hennessy J. *Computer Architecture. A Quantitative Approach. Second Edition. MKP, Inc. 1996.*
3. Texas Instruments, Inc., "TMS320C60 Technical brief: Very long Instruction Word processor, "HOUSTON, TX, 1994.
4. Мельник А.О. *Програмовані процесори обробки сигналів. Львів, 1999.*
5. NEC Video Signal processor. NEC corporation. 1998.
6. Melnyk A. *DSP System Based on Programmable Processor with Scalable Parametrizable Fast Orthogonal Transforms Hardware Core // Proceedings of the XI Conference "Application of Microprocessors in Automatic Control and Measurement". Warsawa, 1998. Vol.1. P.87-98.*
7. D.Chen and J.Rabaey. *A Reconfigurable Multiprocessor IC for Rapid prototyping of real time data path // ISSCC. February, 1992.*
8. A. Yeng and J.Rabaey *A data-driven Architecture for rapid prototyping of high throughput DSP Algorithms // IEEE VLSI signal processing workshop. October, 1992.*
9. Texas Instruments, Inc., *TMS320C80 technical brief: multimedia-video-signal processor (MVP), Houston TX, 1994.*
10. Мельник А.О. *Спеціалізовані комп'ютерні системи реального часу. Львів, 1996.*
11. *VT-5100 High Speed Vector / FFT Processing module. Valley Technologies.www.valeytech.com. 1998.*
12. *BDSP9124 Digital Signal Processor BUTTERFLY DSP. Sharp microelectronic Technology.www.sharpwa.com. 1998.*
13. *Spee-fee.www.nova.stanford.edu.*
14. M.-T.Sun, T.-C.Chen, and A.M.Gottlieb "VLSI implementation of 16x16 discrete cosine transform // IEEE trans.circuits syst. Vol.36. April, 1989. P.610-617.
15. Мельник А.О. *Процесори обробки сигналів. ИППМ НАН України. Львів, 1989.*
16. S.Y.Kung, *VLSI Array processor. Englewood cliffs, NJ: Prenticehall, 1988.*
17. S. Sjolholm. *L.Lindh. VHDL for designers. Prentice Hall Europe, 1997.*