КОМП'ЮТЕРНІ СИСТЕМИ ТА КОМПОНЕНТИ

УДК 004.27

Н. А. Христинець

РЕАЛІЗАЦІЯ БАГАТОПОТОЧНОСТІ НА АРХІТЕКТУРІ МУЛЬТИМЕДІЙНИХ ПРОЦЕСОРІВ NEXPERIA

Луцький національний технічний університет, Луцьк

Анотація. Розглянуто структуру мультипроцесорних систем на архітектурі сучасних мультимедійних процесорів Nexperia з 32-розрядним обчислювальним ядром. Досліджено сфери застосування мультимедійних процесорів, як процесорів загального призначення та їх функції обробки даних. Мультимедійні процесори використовують або функціональні архітектури з обмеженою гнучкістю, але вищою швидкістю та ефективністю, або програмовані архітектури з підвищеною гнучкістю. Проаналізовано архітектуру одного з процесорів Nexperia – TriMedia TM-1300 та наведено схему його основних компонент, принцип роботи центрального арбітражу шини процесора та способи нарощення його швидкодії. Виявлено, що удосконалені процесори загального призначення забезпечують підтримку мультимедіа шляхом включення нових мультимедійних інструкцій і їх паралельного виконання за допомогою підходу співпроцесора SIMD. Вони забезпечують підтримку мультимедіа, включаючи мультимедіа інструкції в набір інструкцій. Замість виконання певних мультимедійних функцій (наприклад стиснення та 3D графіки), мультимедійні процесори надають спеціально створені інструкції для підтримки загальних операцій у обробці відео. Ці інструкції включають підтримку 8-бітних типів даних (пікселів), ефективну адресацію даних і інструкції вводу/виводу. У статті розглянуто можливості програмної реалізації розпаралелення роботи процесорів за допомогою технологій паралельної обробки, яка досягається способом поділу одного виконання завдання на декілька незалежних менших завдань. Запропоновано програмну реалізацію роботи статичним та анонімним методом. Наведені коди програм та результати їх тестування. Доведено, що розбиття різних частин завдання між кількома обчислювальними ресурсами CPU дає змогу скоротити час виконання програми та вдосконалюють потенційну обчислювальну потужність роботи комп'ютерної системи.

Ключові слова: архітектура комп'ютерів, багатопроцесорні системи, мультимедійний процесор, паралельні обчислення, Nexperia, TriMedia.

Abstract. The structure of multiprocessor systems based on the architecture of modern Nexperia multimedia processors with a 32-bit computing core is considered. The fields of application of multimedia processors as general purpose processors and their data processing functions are studied. Multimedia processors use either functional architectures with limited flexibility but higher speed and efficiency, or programmable architectures with increased flexibility. The architecture of one of the Nexperia processors - TriMedia TM-1300 is analyzed and the diagram of its main components, the principle of operation of the central arbitration of the processor bus and methods of increasing its speed are given. Advanced general-purpose processors have been found to provide multimedia support by incorporating new multimedia instructions and executing them in parallel using a SIMD coprocessor approach. They provide multimedia support, including multimedia instructions in the instruction set. Instead of performing specific multimedia functions (such as compression and 3D graphics), multimedia processors provide purpose-built instructions to support general video processing operations. These instructions include support for 8-bit data types (pixels), efficient data addressing, and I/O instructions. The article examines the possibilities of software implementation of the parallelization of processors using parallel processing technologies, which is achieved by dividing one execution of a task into several independent smaller tasks. A software implementation of the work using a static and anonymous method is proposed. The program codes and the results of their testing are given. It has been proven that the division of different parts of the task between several computational resources of the CPU allows to reduce the execution time of the program and improves the potential computing power of the computer system.

Keywords: computer architecture, multiprocessor systems, multimedia processor, parallel computing, Nexperia, TriMedia.
DOI: https://doi.org/10.31649/1999-9941-2022-55-3-59-64.

Вступ

Голландська компанія Nexperia, яка є брендом Philips, щорічно постачає понад 100 мільярдів товарів мікропроцесорної техніки у світі. Ці продукти визнані еталонами ефективності мікропроцесорів завдяки чудовому поєднанню їх розміру, потужності та продуктивності. Невеликий розмір процесора сприяє економії цінної енергії та простору на платі. Крім того, Nexperia – це мікропроцесорна техніка для унікальної групи чипів, які спрощують розробку мультимедійних пристроїв кожного нового покоління. Гнучкі рішення Nexperia допомагають виробникам задовольняти потреби ринку в інноваційних, привабливих продуктах на цільових споживчих і комунікаційних ринках, починаючи від високо-інтегрованих програмованих систем SoC і супутніх мікросхем до них, а також широкого кола системного програмного забезпечення.

Актуальність

Термін «потокова передача» у багатопроцесорній обробці зазвичай використовується для опису безперервних нескінченних потоків даних без початку чи кінця, які забезпечують постійну подачу цих даних. Таку подачу можна використовувати без необхідності попереднього їх завантаження. Потоки даних генеруються усіма типами джерел у різних форматах і обсягах: від програм, мережевих пристроїв і файлів журналу сервера до активності на веб-сайті, банківських транзакцій і даних про місцезнаходження. Усі вони можуть бути агреговані для безперешкодного збору інформації в реальному часі та подальшої аналітики. Тому, програмна реалізація потокової передачі даних є **актуальною** задачею у сучасних багато потокових мультипроцесорних системах.

Мета

В основі процесорів Nexperia лежить архітектура векторного процесора обробки сигналів з довгим словом (VLIW). Архітектура процесорів з кількома обчислювальними пристроями VLIW характеризується тим, що одна інструкція процесора містить кілька операцій, які мають виконуватися паралельно. На час започаткування цієї архітектури стало зрозуміло, що процесорам загального призначення необхідна дуже велика обчислювальна потужність обробки сигналів у реальному масштабі часу. Такі прилади виходили дорогими, їм були потрібні спеціалізовані мікросхеми для підтримки операцій введення-виведення, і вони споживали багато енергії. Тому, **метою** розробки універсального та економного медіа-процесора стало прискорення роботи мультимедійних додатків як в автономному режимі, так і в обчислювальних системах із мікропроцесорами загального призначення. Багатопроцесорність передбачає як локальний, так і віддалений паралелізм, ефективно обходячи глобальне блокування інтерпретатора, використовуючи підпроцеси замість потоків. Завдяки цьому багатопроцесорна обробка дозволяє програмісту повністю використовувати кілька процесорів на одній машині. Такий підхід успішно реалізовується як на Unix, так і на Windows.

Задачі на дослідження

Розглянемо архітектуру одного із процесорів сімейства Nexperia – TriMedia, TM-1300, який має 32-розрядне обчислювальне ядро VLIW/SIMD з вбудованою кеш-пам'яттю команд (32 Кбайт) та даних (16 Кбайт). Він працює з тактовою частотою до 166 МГц і має векторний співпроцесор, що реалізує алгоритм Хоффмана (для обробки MPEG-2). Також у його складі є співпроцесор зображень, що призначений для перетворення колірного простору (YUV/RGB, 32-розрядний контролер зовнішньої пам'яті (до 64 Мбіт) SDRAM) та набір таймерів. Усі вбудовані периферійні пристрої та співпроцесори працюють незалежно від центрального процесора (ЦП) під керуванням DMA5. Інтерфейс PCI/XIO забезпечує зв'язок TM-1300 із шинами персональних комп'ютерів, а також стандартною мікропроцесорною периферією. Інші цифрові інтерфейси процесора: відеовхід та відеовихід ITU-656, звукові вхід (2 канали) та вихід (до 8 каналів) I2 S, звуковий вихід IEC958 (S/PDIF), керуючий порт I2 C, налагоджувальний порт JTAG, а також синхронний послідовний порт для підключення аналогових та цифрових модемів.

На базі приладів родини TriMedia з зазначеною архітектурою було розроблено високоякісні вироби побутового та промислового призначення. Процесор TM-1300 став основою для приладу PNX1300 нової архітектури Nexperia. Основними відмінностями PNX1300 стали знижені напруга живлення та енергоспоживання, збільшені тактові частоти ядра та основного ОЗП (до 200 і 183 МГц відповідно), підтримка 16- та 32-розрядного інтерфейсу SDRAM обсягом до 256 Мбіт (16 Мбіт). Також було спрощено порядок подачі напруги живлення, виправлено помилки в роботі інтерфейсу PCI та програми-завантажувача. Внаслідок цього, конкурентоспроможність приладів підвищилася. Програми, розроблені для процесорів TriMedia, Nexperia необхідно перекомпільовувати, хоча сумісність на рівні вихідних кодів і бібліотек API гарантована.

Розв'язання задач

Головною ідеєю, реалізованою в сімействі процесорів Nexperia, є гнучка обчислювальна система, що має самостійну периферію і програмно адаптується до виконуваного додатку. Мета впровадження нової архітектури – максимально збільшити швидкістю обчислень. Крім того, чим точніше обчислювальний пристрій може бути налаштований на клас додатків, тим більше його ефективність (прівняно з мікропроцесорами ЦПОС загального призначення). Система, реалізована на основі Nexperia, працює під управлінням компактного ядра операційної системи реального часу (pSOS), що виконується центральним процесором VLIW. Ключові моменти в роботі даної архітектури – поділ у часі роботи ЦП та периферії та «спілкування» всіх пристроїв через ОЗП за допомогою шини та механізму DMA.

Центральний процесор TM32 (рис. 1) перемикається від одного завдання до іншого наступним чином. Спочатку він декодує відеокадр, потім кілька пакетів аудіопотоку, повертається до відеоданих, керує портом PCI, залучає графічний співпроцесор для ресурсомісткої операції масштабування зображення тощо. За кожний такт ЦП може задіяти до 5 пристроїв одночасно, включаючи периферію та 27 конвеєризованих функціональних блоків.

Для успішного виконання функцій «диригента» центральний процесор звільнений від деяких операцій, характерних для мікропроцесорів загального призначення, наприклад, трансляції адрес або обчислень із подвійною точністю; з нього виключено планувальник команд і суперскалярну логіку та тим самим знижено апаратні витрати. Проте, набір команд TM32 включає усі стандартні операції. Внутрішня шина процесорів Nexperia також забезпечує доступ до регістрів стану та управління всіх пристроїв та зовнішньої периферії. Вона складається з двох окремих 32-розрядних шин адреси та даних. Для обміну використовується протокол групового пересилання.

Доступом до шини керує центральний арбітр, до якого підведено лінії запитів від кожного пристрою, що має можливість захоплення шини (bus master). Алгоритми роботи арбітра шини можуть залежати від виконуваної програми, як і смуга пропускання шини, яка виділяється пристроям. Кожен режим

роботи центрального арбітра гарантує мінімальну смугу пропускання та максимальний час очікування конкретного пристрою.

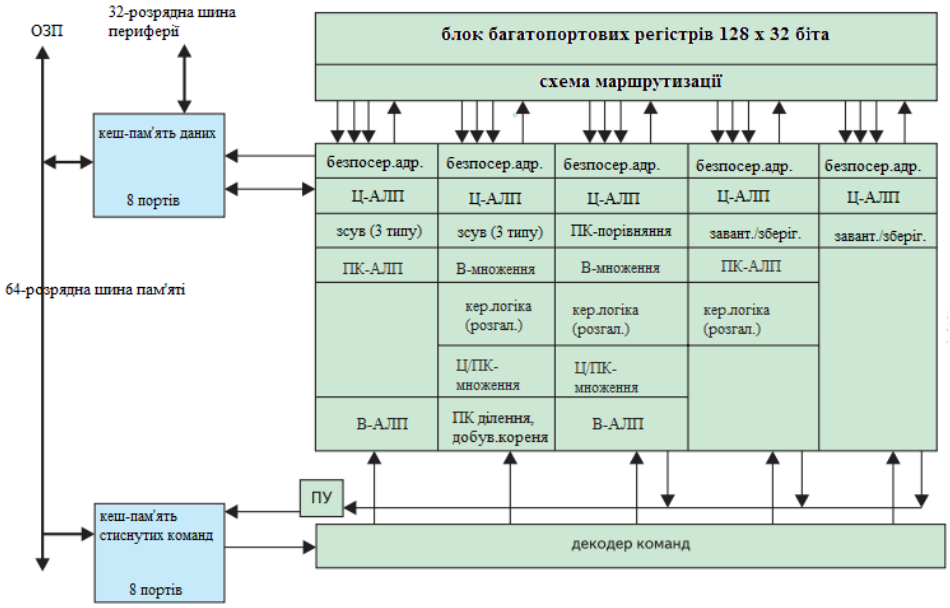

Рисунок 1 – Спрощена блок схема центрального процесора TriMedia (TM32). Умовні позначення: Ц − цілочисельний, ПК − плаваюча кома, В − векторний, ПУ − програмне управління

Описаний принцип роботи є однією з «родзинок» процесорів TriMedia/Nexperia, що забезпечують прискорення роботи мультимедійних програм. Для порівняння, перехресний комутатор (crossbar switch), що використовується у VLIW-процесорах компанії Texas Instruments, не підтримує гнучку систему пріоритетів пристроїв, його простий механізм вирішення конфліктів при запиті доступу до шини працює за жорстко обумовленою схемою. В цьому випадку необхідна швидкодія системи «добирається» необхідною кількістю АЛП, розвиненим кешуванням, потужністю співпроцесорів та високою тактовою частотою (до 1 ГГц).

Гнучкість процесорів Nexperia, їх унікальна здатність тонкого налаштування на виконуваний додаток, пред'являють особливі вимоги до інструментального програмного забезпечення. У структурі вищезгаданого процесора, планувальник команд інтегрований із розподільником регістрів, він підтримує захищене виконання команд, конвеєрну організацію та керування функціональними блоками ЦП та системи в цілому. Програмна реалізація багатопоточності дозволяє наочно продемонструвати ефективність розпаралелювання на рівні команд та зниження кількості звернень до регістрів та ОЗП.

Реалізуємо програмно кілька методів паралельного використання потоків через пасивне очікування, яке реалізується за допомогою операційної системи. Операційна система зберігає контекст потоку та вивантажує його, надаючи можливість виконуватись іншим потокам. Досліджено, що завантаження процесора при активному очікуванні в середньому дорівнює 92%. При пасивному очікуванні основний потік вивантажується і зайнятість ЦП в середньому дорівнює 50%.

Статичний метод дозволяє обробляти паралельно довільну кількість потоків незалежно один від одного (тут є суттєва різниця з статичним класом, доступ до якого необхідно організовувати потокобезпечним). У наведеному фрагменті паралельно виконуються первинний та вторинний (дочірній) потоки:

```
// Статичний метод, який планується виконувати одночасно в головному (первинному) та у вторинних потоках.
    static void WriteSecond()
    {
      // CLR призначає кожному потоку свій стек із власними локальними змінними.
      // Окремий екземпляр змінної counter створюється в стеці кожного потоку,
      // тому для кожного потоку виводяться свої значення counter
      int counter = 0;

      while (counter < 10)
```

```
        {
            Console.WriteLine("Thread Id {0}, counter = {1}", Thread.CurrentThread.GetHashCode(), counter);
            counter++;
        }
    }

    static void Main()
    {
        // Робота вторинного потоку.
        Thread thread = new Thread(WriteSecond);
        thread.Start();

        // Робота первинного потоку.
                                                    WriteSecond();
```

Лістинг ефекту операцій за даним методом наведено на рисунку 2.

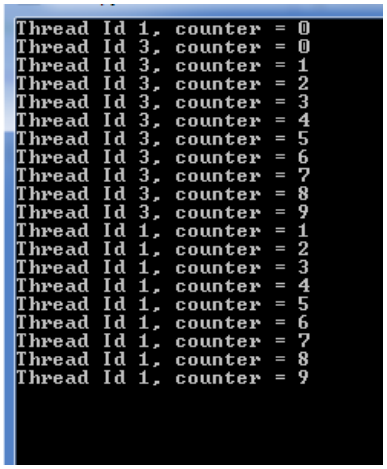

Рисунок 2 – Лістинг ефекту операцій за статичним методом

Наступний, *анонімний метод* створює безіменний блок коду, пов'язайни з конкретним екземпляром делегата. Для створення анонімного методу зазначено кодовий блок після ключового слова delegate. У наведеному прикладі анонімний метод служить виведенням повідомлень «№ counter = 1» від 1 до 3. Спочатку у програмі оголошується тип делегата CountIt без параметрів і з типом void. Далі у методі Main() створюється екземпляр count делегата, якому передається кодовий блок, наступний після ключового слова delegate. Саме цей кодовий блок і є анонімним методом, який виконуватиметься при зверненні до делегата count. В програмі використано скрочений синтаксис анонімного методу. Потік має властивість IsAlive, що повертає true після виклику Start() і до завершення потоку. Потік, який закінчив виконання, не може бути розпочато знову:

```
// Потоки. Анонімні методи.

namespace Threads
{
    class Program
    {
        static void Main()
        {
            int counter = 0;

            // Початок потоку
            Thread thread = new Thread(delegate() { Console.WriteLine("1. counter = {0}", ++counter); });
            thread.Start();

            Thread.Sleep(100);
            Console.WriteLine("2. counter = {0}", counter);
```

```
            // Параметри
            thread = new Thread((object argument) => { Console.WriteLine("3. counter = {0}", (int)argument); });
            thread.Start(counter);

            // Затримка
            Console.ReadKey();
        }
    }
}
```

Результати тестування даного методу наведені на рисунку 3.

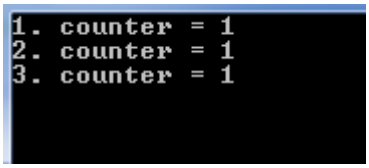

Рисунок 3 – Результати тестування анонімного методу

Наведені методи демонструють паралельне виконання коду через багатопоточність. Використання кількох потоків виконання – це один із способів забезпечити можливість реагування програми на дії користувача при одночасному використанні процесора для виконання завдань між появою або навіть під час появи подій користувача.

Висновки

Існує безліч засобів для оцінки продуктивності комп'ютерних систем. Деякі тести для такої оцінки вимірюють час виконання часто уживаних алгоритмів, інші оцінюють виконання завдань, специфічних для конкретної прикладної області, треті формують власний синтетичний показник, заснований на вимірі продуктивності різних комп'ютерних підсистем. Сучасні архітектури багатопроцесорних обчислювальних систем настільки різноманітні і складні, що при проектуванні нових алгоритмів необхідно враховувати значну кількість факторів. Ігнорування приватних архітектурних особливостей, закладених розробниками у обчислювальну систему, може призвести до значного падіння продуктивності та ефективності прикладної програми. Розглянуто теоретичну сутність мультипроцесорів Nexperia та виявлено, що мультипроцесорна архітектура забезпечує суттєве підвищення надійності системи, можливості розпаралелення інформаційних потоків. Програмні реалізації методів паралельних обчислень передбачають виконання їх на системі з двома процесорами, один з яких може працювати в режимі ядра, а інший – в режимі користувача. В подальшому ці алгоритми можна буде допрацювати, реалізувати на іншій мові програмування та застосувати для ряду інших прикладних задач галузі інформаційних технологій.

Список літератури

[1] L. Wanga, “Research on the Performance of Robot Multiprocessor Control System Based on BS Structure Digital Media”, *Microprocessors and Microsystems*. 2020. [Електронний ресурс]. Режим доступу: https://www.sciencedirect.com/science/article/abs/pii/S0141933120300910#. Дата звернення: 20 серпня 2022.

[2] В. О. Денисюк, С. М. Цирульник, *Мікропроцесорні системи управління: навч. посіб.* Вінниця, Вінн. нац. аграр. ун-т: ТВОРИ, 2021, 204 с.

[3] V. Padmajothi, J. MazherIqbal, V. Ponnusamy “Load - aware intelligent multiprocessor scheduler for time-critical cyber-physical system applications”, Computers & Electrical Engineering. 2022. [Електронний ресурс]. Режим доступу: https://www.sciencedirect.com/science/article/pii/S0045790621005462#. Дата звернення: 20 серпня 2022.

[4] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski and C. Kozyrakis, “Evaluating MapReduce for Multi-core and Multiprocessor Systems”, 2007 IEEE 13th International Symposium on High Performance Computer Architecture, 2007, pp. 13-24.

[5] Сайт компанії «Nexperia». [Електронний ресурс]. Режим доступу: https://www.nexperia.com. Дата звернення: 20 серпня 2022.

Стаття надійшла: 10.09.2022.

References

[1] L. Wanga, “Research on the Performance of Robot Multiprocessor Control System Based on BS Structure Digital Media”, *Microprocessors and Microsystems*. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0141933120300910#. Accessed on: August 20, 2022.

[2] V. O. Denisyuk, S. M. Tsirulnyk, *Microprocessor control systems: academic. Manual*. Vinnytsia, Vinn. national agrarian university: CREATIONS. 2021, 204 p.
[3] V. Padmajothi, J. MazherIqbal, V. Ponnusamy "Load - aware intelligent multiprocessor scheduler for time-critical cyber-physical system applications", Computers & Electrical Engineering. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0045790621005462#. Accessed on: August 20, 2022.
[4] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski and C. Kozyrakis, "Evaluating MapReduce for Multi-core and Multiprocessor Systems", 2007 IEEE 13th International Symposium on High Performance Computer Architecture, 2007, pp. 13-24.
[5] Nexperia website. [Online]. Available: https://www.nexperia.com/. Accessed on: August 20, 2022.

Відомості про авторів

Христинець Наталія Анатоліївна – кандидат технічних наук, старший викладач кафедри комп'ютерної інженерії та кібербезпеки.

N. A. Khrystynets

IMPLEMENTATION OF MULTI-THREADING ON THE ARCHITECTURE OF NEXPERIA MULTIMEDIA PROCESSORS

Lutsk National Technical University, Lutsk