УДК 621.382

ЯДРО З АПАРАТНИМ РОЗПАРАЛЕЛЮВАННЯМ ВИКОНАННЯ ПРОГРАМ

М. В. Стасюк, О. І. Антонюк, Д. Ю. Лебедев

Національний технічний університет України «Київський політехнічний інтитут»

int2080@ukr.net

Описано ядро для мікроконтролера, яке апаратно псевдо розпаралелює виконання кількох підпрограм. Проаналізовано перевагу такої реалізації над звичним програмним перемиканням між підпрограмами.

Ключові слова: ядро мікроконтролера, переривання, арифметичний пристрій, мультиплексування.

This article describes the core of the microcontroller, which on hardware pseudo parallels the perform several routines. The advantages of such an implementation over conventional software switching between routines.

Keywords: microcontroller core; break; arithmetic unit; multiplexing.

Вступ

Під час розв'язання задач в електроніці постійно виникає необхідність виконувати кілька підпрограм одночасно.

Наприклад, обслуговувати кілька переривань або виконувати програми користувача у випадку роботи ОС реального часу. Сьогодні це завдання вирішується за допомогою виділення відрізків часу, протягом якого ядро виконує одну підпрограму, або введення кількох ядер. Останній варіант вирішення суттєво ускладнює алгоритми розподілення задач між ядрами і збільшує необхідні апаратні ресурси для реалізації.

Мета роботи — створення ядра, яке виконуватиме інструкції кожної підпрограми по черзі. Таким чином, кількість апаратних ресурсів суттєво не збільшується, але виконання інструкцій є псевдо паралельним.

Такий підхід дозволяє економити такти на перемикання контексту; обробник переривання починає виконуватись майже відразу після надходження переривання хоч і на менш ефективній частоті, що дає змогу виконати скидання прапорів чи збереження результату роботи периферії швидше, ніж це робиться зазвичай.

Ядро призначене для використання у мікроконтролерах. Ядра, спроектовані для процесорів, мають набагато складнішу будову і закладені алгоритми. За це доводиться платити ціною розробки і площею на кристалі. Поки що розпаралелювання досягається двома шляхами [1], [2]:

- векторно-конвеєрним;
- масивно-паралельним.

Векторно-конвеєрний спосіб (SIMD машини, single instruction multiply data) передбачає виконання одного потоку команд над кількома потоками даних.

Масивно-паралельний спосіб (MIMD машини, multiply instructions multiply data) використання кількох ядер в одній системі.

У першому випадку виграш іде за рахунок економії тактів для декодування команди. У другому випадку — за рахунок кількості ядер, а отже команд, які виконуються паралельно.

Як зазначалось, запропонований спосіб розпаралелювання дає виграш за рахунок відсутності потреби перезавантажувати конвеєр у разі команди переходу і часу на перемикання контексту. Програми виконуються ніби на різних ядрах, тому запропонований спосіб нагадує масивно-паралельний, хоч і реальної паралельності не існує.

Подібне ядро ще не було реалізоване в жодному контролері. Мікроконтролери розвиваються сьогодні в кількох напрямках [3]:

– перехід до 32-бітних обчислень;
– розширення набору команд;
– збільшення тактової частоти;
– збільшення периферії;
– зменшення площі на кристалі (здешевлення).

Перехід до паралельної обробки не використовується.

Структурна схема ядра

Ядро містить один конвеєр, що дозволяє пришвидшувати роботу за рахунок відсутності частин, що нічим не зайняті.

Команди переходів зазвичай створюють необхідність перезавантажувати конвеєр, оскільки команда, яка декодується в декодері, вже не повинна виконуватись. У розробленому ядрі перезавантаження конвеєра, викликане командами розгалуження, не відбувається, оскільки конвеєр буде містити команди інших підпрограм. В ядрі використано команди різної довжини, що дозволить економити пам'ять. Набір команд подібний до команд ядра AVR з певними змінами. Адреси мають ширину 16 біт, дані — 8 біт.

Ядро складається з частин, зображених на рис. 1.

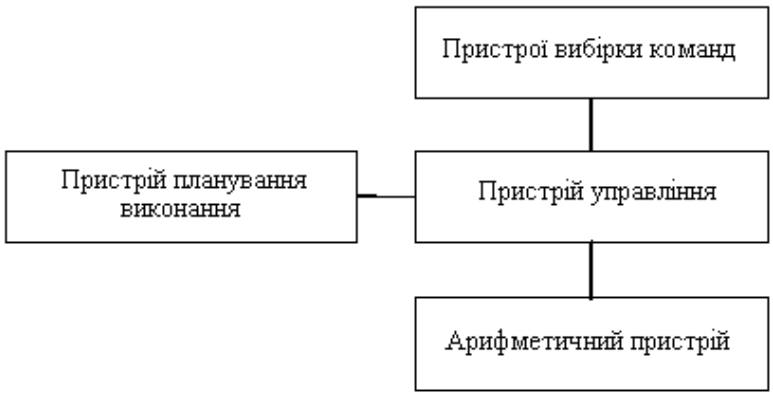

Рис. 1. Структурна схема ядра

Пристрій управління — блок, який реалізує команди розгалуження, пересилання даних, звернення до внутрішніх регістрів, перехід до виконання обробників переривання. Внутрішні регістри — це вказівники команд (PC — *program counter*), вказівник вершини стеку (SP — *stack pointer*), прапори (*flags*), регістр контролю виконання (FCR — f*low control register*), регістр дозволу виконання потоків (TER — *thread enable register*). Арифметичний пристрій виконує арифметичні команди, логічні, бітові, команди зміни прапорів. Пристрої вибірки команд реалізують функції міні кешу команд. Необхідність їх реалізації викликана різною довжиною команд. При конвеєрному виконанні вибірка наступної команди виконується одночасно з декодуванням попередньої, тобто коли ще невідомо на скільки треба збільшити PC. Пристрій планування виконання керує послідовністю виконання потоків. Усього реалізовано чотири блоки регістрів, які зберігають контекст підпрограми, тобто можна запустити чотири потоки виконання. Кожен потік має однаковий пріоритет з іншими, активні потоки виконуються по черзі. Неактивні не виконуються взагалі. Активний/неактивний стан установлюється бітами в TER. Пристрій управління агрегує інші пристрої.

Для розуміння роботи ядра розглянемо кодування команд. Як уже зазначалось, команди мають різну довжину. Першим байт містить код операції: біти 0–5; і розмір команди: біти 6–7. Таким чином розмір команди можна отримати, повністю не декодуючи її. Необхідність такого прийому викликана знову ж таки різною довжиною команд. Другий байт — з індексами регістрів-операндів або зміщення для умовних переходів. Другим і третім байтом може бути адреса для переходу або виклику підпрограми. Третім і четвертим байтом може бути адреса пам’яті даних для команди прямого завантаження в регістр/запису в пам’ять. Типи команд наведено на рис. 2.

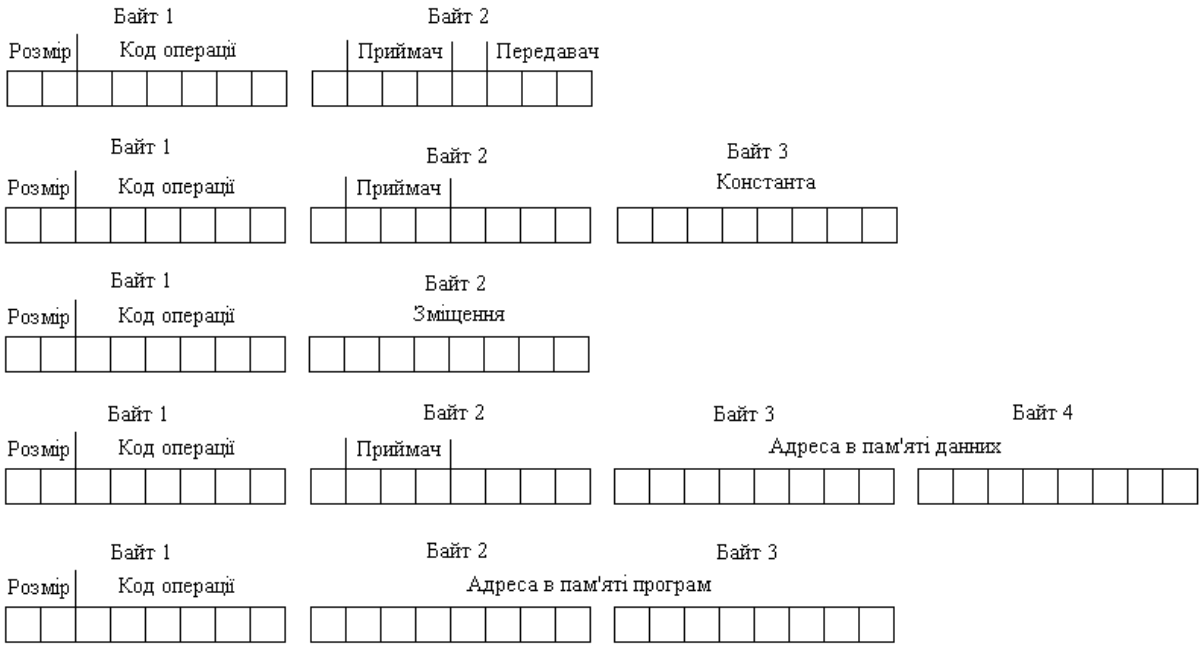

Рис. 2. Типи команд

Арифметичний пристрій має просту реалізацію, тому структурної схеми наводити не будемо. Він являє собою комбінаційну схему, результати на виході якої формуються під впливом вхідних даних і коду інструкції. Декодування інструкцій, які виконує арифметичний пристрій, відбувається відразу в АЛП. Список команд АЛП: додавання регістрів, додавання з прапором перенесення, додавання з константою, віднімання регістрів, віднімання з прапором перенесення, віднімання константанти, обернене значення, інкремент, декремент, порівняння регістрів, порівняння з константою, логічне І, АБО, ВИКЛЮЧНЕ АБО регістрів і регістра з константою, побітна інверсія, зсуви, циклічні зсуви, зсуви через прапор перенесення вліво і вправо, арифметичний зсув вправо, обмін тетрад, очищення, встановлення біта, збереження біта у прапора Т і навпаки, установка/очищення прапорів С, Т, І, PP, ATM. У набір прапорів результату виконання входять прапори:

- перенесення;
- переповнення;
- знака;
- нуля;
- прапор користувача;
- парності.

Опис прапорів управління потоком буде наведено нижче.

Пристрій вибірки команд складається з кільцевого буфера і внутрішнього регістра з адресою для вибірки, яка не обов'язково вказує на початок команди. Розмір буфера 8 байт — розмір 2-х найдовших команд. На вхід пристрою подається три сигнали:

– встановити адресу — пристрій виставляє адресу, яка знаходиться у внутрішньому вказівнику адрес або поточну адресу одного з РС. РС виставляється, якщо була зміна в РС у попередньому такті (команда розгалуження або програмна конфігурація РС іншим потоком), а у поточному такті виставлено сигнал «встановити адресу». В цьому самому такті у внутрішній вказівник команд записується значення РС+4 (у буфері прочитається чотири байти навіть, якщо команда матиме меншу довжину). Якщо зміна РС була, а сигнал «встановити адресу» відсутній, то у внутрішній вказівник адрес збережеться просто значення РС;

– прочитати команду — пристрій читає команду, яка виставлена на шині команд. Якщо в буфер не поміщається усі чотири байти — буде збережено менше. Внутрішній вказівник адреси на відповідну величину;

– інкремент — сигнал виставляється в циклі виконання команди, яка збережена в буфері. Вказівник на початок команди в кільцевому буфері збільшується на розмір команди.

Оскільки ядро виконує чотири потоки, то і кількість пристроїв вибірки також чотири — по одному на потік. Шини з командами від всіх пристроїв вибірки мультиплексуються в одну шину, яка йде до пристрою управління і АЛП.

Пристрій планування виконання слугує для формування послідовності виконання інструкцій кожного потоку. На його роботу впливають прапори дозволу виконання потоку і прапори управління потоком. Прапори дозволу виконання потоку знаходяться в регістрі TER. «1» у відповідному біті запускає потік на виконання, тому перед запуском потрібно проініціалізувати відповідний РС. Прапори управління потоком:

– глобальний дозвіл переривань (GIE);

– дозвіл паралельного виконання потоків (PPE) — при зніманні прапора, буде виконуватись підпрограма одного потоку, хоч інші і можуть бути ввімкнені;

– дозвіл паралельного виконання обробника переривання із звичайною підпрограмою (SIM) — «1» у прапорі дозволяє паралельно виконувати обробник переривань із звичайними підпрограмами. Інакше всі підпрограми будуть призупинені, доки буде активний принаймні один обробник переривань. Паралельністю виконання обробників у цьому випадку керує прапор PPE;

– критична секція (ATM) — атомарне виконання інструкцій. Маскуються всі переривання і зупиняються всі потоки, крім того, що надійшов у критичну секцію.

Цей блок також ставить у чергу потоки, які не дозволені, але надійшов запит на переривання, який має обслуговувати цей потік. Детальніше про систему переривань написано нижче.

Структурна схема пристрою управління наведена на рис. 3.

Пристрій управління є з'єднувальною ланкою всього ядра. В ньому виконуються команди розгалуження, запису у внутрішні регістри, пересилання даних. Система переривань дає можливість конфігурувати пріоритети переривань і номер потоку, в якому буде виконуватись переривання. Ці налаштування повинні бути реалізовані в контролері. Ядро має чотири адресні лінії, на яких буде виставлятись адреса обробника переривання, чотири лінії запиту на перехід до обробника переривання і чотири лінії підтвердження переходу на обробник переривання. При виникненні переривання контролер повинен виставити адресу, на одну із ліній запиту і адресу на адресну лінію з таким самим номером. Номер лінії означає номер потоку, в якому буде виконуватись обробник.

Рис. 3. Структурна схема пристрою управління

Пристрій управління потоком виконання поставить цей потік у чергу навіть, якщо він вимкнений у регістрі TER. Відбудеться вибірка команди (необов'язково коректної, оскільки блок вибірки може бути не сконфігурований), декодування вибраної команди, але в такті виконання відбудеться виклик процедури обробника. Виконання обробника не може бути перерване іншим обробником.

Висновки

Ядро має набір команд, який нічим не поступається наборам існуючих ядер, тому може використовуватись у тих самих галузях, що й інші ядра. Швидкість виконання в 1 MIPS на 1 МГц і наявність розпаралелювання дає можливість виконувати на ньому складні програми критичні до часу виконання. Показано новий шлях розвитку мікроконтролерів.

Подану архітектуру можна суттєво ускладнювати за рахунок збільшення кількості потоків, які виконуються паралельно, додавання нових команд і обчислювальних блоків.

ЛІТЕРАТУРА

1. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем / Дж. Ортега. — М. : Мир, 1991. — 365 с.

2. *Векторизация* программ: теория, методы, реализация: сборник / под ред. И. Н. Теплюка. — М. : Мир, 1991. — 272 с.

3. *Павлов П.* Тенденции развития микропроцессоров и микроконтроллеров / П. Павлов // Современная электроника. — 2007. — № 2.

REFERENCES

1. *Ortega J.* Introduction to parallel and vector methods for solving linear systems / J. Ortega. — M. : Mir, 1991. — 365 p.

2. *Vectorized* programs: theory, methods, realization: Sat / ed. I. N. Teplyuk. — M. : Mir, 1991. — 272 p.

3. *Pavlov P.* Trends of microprocessors and microcontrollers development / P. Pavlov // Modern electronics. — 2007. — № 2.

Стаття надійшла до редакції 10.12.2013