

випадку $D_{x.2.0} = D_{x.1.8}$. Цей же сигнал через час Θ_d (діаграма 2), необхідний для обчислення кореляційної функції на інтервалі $\Theta_{1.8}$, переводє реєстр RG у режим виводу даних, після чого обнулить блок BK .

На інтервалі Θ_2 процес локально-стаціонарний протягом $4\Theta_d$ і нестаціонарний на інтервалі $5\Theta_d$, бо $\Delta D_x = D_{x.2.5} - D_{x.2.0} > E_{D_x}$. Тому на виході BA виробляється сигнал «3», який, пройшовши через елемент «АБО» (діаграма 2), переводє RG у режим виводу, після чого обнулює BK . Крім того, вихідний сигнал з BA , пройшовши через двохходовий елемент «АБО» (діаграма 3), сприяє записові в пам'ять $\Theta_{2.5}$ значення $\Theta_{3.0}$. При цьому вихідний сигнал BA обнулює також DL_x і DL_y , оскільки вони в даний момент містять значення нестационарного процесу на інтервалі $\Theta_{2.5}$, у зв'язку з чим оцінка кореляційної функції не обчислюється.

Обчислення оцінок кореляційної функції на інтервалі Θ_3 починається з інтервалу $\Theta_{3.1}$. На інтервалі Θ_3 процес локально-стаціонарний протягом Θ_d і нестаціонарний на інтервалі $2\Theta_d$. За сигналом BA у момент T_4 виводиться обчислена протягом інтервалу $\Theta_{3.1}$ оцінка кореляційної функції, а також значення $\Theta_3 = \Theta_d$. Блоки BK , DL_x і DL_y обнулюються й оцінки кореляційних функцій на інтервалі $\Theta_{3.2}$ не обчислюються.

На інтервалі $\Theta_4 = \Theta_d$ процес нестационарний. Сигнал «1» обнулює BK ще до приходу на нього затриманого в DL_x і DL_y оцифрованого процесу, обнулює DL_x і DL_y та закладає в пам'ять BA оцінку $D_{x.4.0}$.

На підставі інших оцінок (M_x , M_y , D_y , D_{xy}) робота системи здійснюється аналогічно.

Отже, розглянуті методи та цифрова апаратура дають змогу в режимі реального часу автоматично вибирати оптимальні інтервали аналізу деяких класів НВП.

1. Воллернер Н. Ф. Аппаратурный спектральный анализ сигналов. М., 1977. 2. Погрибной В. А. Бортовые системы обработки сигналов. К., 1984.

Стаття надійшла до редакції 29.04.91

УДК 621.3:621.372.54

Б. І. ЯВОРСЬКИЙ

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ АЛГОРИТМІВ ЦИФРОВОЇ РЕКУРСИВНОЇ ФІЛЬТРАЦІЇ

При цифровій рекурсивній фільтрації (ЦРФ) сигналів у процесорі над кодами відліків сигналу проводяться операції множення на постійні коефіцієнти, сумування й обміну типу

© Яворський Б. І., 1993

регістр—регістр, пам'ять—регістр, пам'ть—пам'ять [1]. Алгоритм ЦРФ ефективний, коли габарити, споживана потужність, вартість процесора, що його виконує, при достатній надійності функціонування в реальному часі задовольняють вибрані критерії. Критеріями ефективності можуть бути, наприклад, затрати часу чи обладнання, необхідні для обробки одного відліку сигналу. Пошук ефективних алгоритмів здійснюється шляхом аналізу їх топологічних моделей (цифрових кіл), отриманих при тотожних перетвореннях [1].

Топологічні моделі можна розглядати як інтерпретацію логіко-алгебраїчних моделей ЦРФ, до яких входять множини операцій, результатів операцій, моментів часу їх отримання, а також відношення послідовності та паралельності [2, 3] (порядку й еквівалентності [4]). Разом з топологічними логіко-алгебраїчні моделі відбивають просторово-часові властивості та характеристики процесора й алгоритму. При побудові результативного алгоритму використовують відношення порядку. Відношення еквівалентності застосовують при розпаралелюванні операцій, що сприяє підвищенню швидкодії (за рахунок збільшення кількості обладнання) [1].

При розпаралелюванні розглядають їх арифметичні множини. Проте при пошуку ефективних алгоритмів слід розглядати також відношення, притаманні множинам операцій обміну. Крім того, при встановленні відношень потрібно враховувати топологію та логіко-алгебраїчні характеристики конкретного процесора, що виконуватиме даний алгоритм ЦРФ. Це дає змогу підвищити ефективність ЦРФ. При пошуку ефективного алгоритму ЦРФ ми виходимо із конкретної топології процесора, що виконує алгоритм, і розглядаємо відношення еквівалентності на множині всіх операцій, в тому числі операцій обміну.

ЦРФ реалізують послідовним з'єднанням ланок не вище другого порядку [5]:

$$\begin{cases} y_{n,j} = g_{2,j} y_{n-2,j} + g_{1,j} y_{n-1,j} + x_{n,j}, \\ y_{вих,j} = g_{0,j} (y_{n-2,j} - y_{n,j}), \end{cases} \quad (1)$$

де j — номер ланки; $j = \overline{1, N}$, N — кількість ланок; $g_{0,j}$, $g_{1,j}$, $g_{2,j}$ — постійні коефіцієнти; $x_{n,j}$ — відлік вхідного сигналу для $j=1$ і $x_{n,j}=0$ — для $j>1$; $y_{вих,j}$ — відлік вихідного сигналу.

Нехай процесор, що складається з надоперативної регістрової пам'яті R , оперативної пам'яті Y та постійної пам'яті коефіцієнтів C , виконує операції типу сумування $+$, $-$, множення \times , присвоєння (обміну) $=$ і реалізує регістрову та відносно індексну адресації. Тоді, встановивши відношення порядку на множині $\{Y, R\}$ результатів операцій $+$, $-$, \times для (1), можна дійти до алгоритму, який зображений підпрограмою *FILTR1* (колонка 1) на ФОРТРАНІ. Тут множиною результатів операцій є $\{R0, R1, Y(J), Y(J+1)\}$. Відношення порядку на множині $\{Y, R\}$ вимагає такого розміщення елементів цієї множини, щоб алгоритм став результативним (результат обчислень — корект-

ним). На кроках 4, 6, 8 і 5, 7 обчислюють значення елементів R_0 , $Y(J)$ множини $\{Y, R\}$. Це є результатом відношення еквівалентності та вказує на можливість паралельного виконання операцій. Новий алгоритм зображений підпрограмою *FILTR2* (колонка 2). Перехід від підпрограми *FILTR1* до підпрограми *FILTR2* завдяки синтаксису мови ФОРТРАН може здаватися очевидним. При мікропрограмуванні процесора такий перехід не є очевидним.

Розглянемо тепер операції обміну між комірками пам'яті та регістрами (кроки 6, 7 підпрограми *FILTR2*). На множині комірок $\{Y(J), Y(J+1), J=1, N\}$ кожної пари $n, n+1$ номерів відліків існує відношення еквівалентності. Наприклад, при обчисленні значення $n+1$ відліку $y_{вих, N}$ замість $Y(J+1)$ слід використовувати $Y(J)$, а замість $Y(J)$ — вміст комірки $Y(J+1)$, помістивши туди попередньо обчислене R_1 .

Такий алгоритм зображений підпрограмою *FILTR3* (колонка 3). При використанні спеціалізованого процесора ЦРФ необхідно організувати в ньому сторінково-відносну адресацію, по чергово змінюючи при кожному новому n сторінку пам'яті. Апаратно все зводиться до використання *RS*-тригера для формування адреси (у підпрограмі *FILTR3* він імітується на 4—7 кроках). У результаті врахування еквівалентності комірок пам'яті вдається уникнути операції обміну між комірками пам'яті, що сприяє збільшенню швидкодії.

Виявити відповідні відношення можна при наявності певного формального апарата. У даному випадку для цього використовували матричне зображення логіко-алгебраїчної моделі ЦРФ при послідовному з'єднанні ланок другого порядку [2]. Наявність відповідного відношення встановлювали за властивостями (вигляду) матриці (див., наприклад, [4]).

Отже, при аналізі алгоритмів ЦРФ для вибору найбільш ефективного з них пропонуємо таку методику:

а) вибрати базовий алгоритм; б) встановити відношення порядку для отримання результативного варіанту алгоритму; в) не порушуючи відношення порядку, проводити тотожні перетворення та розглядати відношення еквівалентності на всіх результатах операцій і обчислювати вибрані критерії ефективності отримуваних варіантів алгоритмів; г) вибрати варіант зі значенням критерію, яке задовольняє задані умови.

Базовий алгоритм вибирають на основі розв'язку задачі синтезу та рекомендацій, пов'язаних з реалізацією алгоритму (стійкістю, чутливістю до малих змін коефіцієнтів і шумами квантування).

1. Крошьер, Опенгейм. Анализ линейных цифровых цепей // ТИИЭР. 1975. Т. 63. № 4. С. 45—46. 2. Петренко А. И., Артюхов А. Г., Кондратюк В. А., Подладчиков В. И. Матричные модели цифровых цепей // Автоматизация проектирования в электронике. К., 1979. Вып. 19. С. 6—17. 3. Свяцкий А. Б., Шаповаленко С. В., Шмид А. В. Кремниевые компляторы СБИС и их применение в САПР ЭВМ // Итоги науки и техники. Сер. Техническая кибернетика. М., 1979. Т. 27. С. 39—99. 4. Сигорский В. П. Математический аппарат инженера. К., 1975. 5. Яворский Б. И., Домбров-

УДК 621.376.56

О. В. ТИМЧЕНКО

КОДЕРИ З ЛІНІЙНОЮ ТА БАГАТОРІВНЕВОЮ ДЕЛЬТА-МОДУЛЯЦІЄЮ ДЛЯ ЦИФРОВОЇ ФІЛЬТРАЦІЇ НА ООС

Використання однорідних обчислювальних середовищ (ООС) дає змогу за рахунок розпаралелювання обробки по швидкодіючих однокітових процесорах різко збільшити продуктивність цифрової фільтрації (ЦФ). Витрати ООС-поля пропорційні до ступеня розпаралелювання обробки і збільшуються зі зростанням розрядності оброблюваних даних. Тому актуальним є зменшення розрядності сигналів. Для збереження заданої роздільної здатності ЦФ використовують альтернативну форму подачі сигналів з низькою розрядністю — у форматі дельта-модуляції (ДМ), причому широко застосовують як однорозрядну лінійну ДМ (ЛДМ), що найбільше підходить до використання в однорозрядних ООС, так і адаптивні види ДМ — багаторівнева ДМ (*multilevel* ДМ—МДМ) і диференціальна імпульсно-кодова модуляція (ДІКМ) [2]. Зменшення розрядності сигналів досягається у цих випадках за рахунок того, що квантуванню підлягає не сам сигнал $x(t)$, а похибка апроксимації $a(t) = x(t) - \hat{x}(t)$, що відповідає його приросту в моменти дискретизації $t = kT$, де $k > 0$; $\hat{x}(t)$ — сигнал апроксимації; T — період дискретизації.

На виходах дельта-кодера бажано формувати як сигнал ЛДМ (МДМ), так і ІКМ, тому що одночасне їх застосування дає змогу спростити алгоритм обробки [3], а значить, додатково скоротити об'єм ООС-фільтра.

Розглянемо алгоритми функціонування та структури ЛДМ-кодерів, за допомогою яких обробляються вхідні сигнали в широких динамічному і частотних діапазонах з можливістю одночасного формування ЛДМ-, МДМ- і ІКМ-кодів.

Кроки квантування ЛДМ-кодерів формуються за рівнянням [4]

$$e_i^{(x)} = \begin{cases} \varepsilon^{(x)}, & \alpha(kT) \geq 0; \\ -\varepsilon^{(x)}, & \alpha(kT) < 0, \end{cases} \quad (1)$$

де $\varepsilon^{(x)}$ — крок квантування при ЛДМ, сигнал апроксимації