УДК 004.3

ЖАБИН В.И.,
МАКАРОВ В.В.

ИСПОЛЬЗОВАНИЕ ТАБЛИЦ ФУНКЦИЙ ДЛЯ БЫСТРОГО ВЫЧИСЛЕНИЯ МНОГОЧЛЕНОВ

Предложен таблично-алгоритмический метод вычисления многочленов, основанный на предварительной обработке коэффициентов. Показана возможность ускорения вычисления многочленов по сравнению с реализацией известных таблично-алгоритмических методов.

The table and algorithmic method of calculation of polynomials based on preliminary coefficient processing is offered. Possibility of acceleration of calculation of polynomials in comparison with realization of the well-known table and algorithmic methods is shown.

Введение

Непрерывное расширение области применения цифровой вычислительной техники приводит к необходимости решения все более сложных задач обработки информации, что, в свою очередь, требует постоянного повышения эффективности аппаратных средств и программного обеспечения вычислительных систем. Одним из основных направлений ускорения процессов обработки информации является аппаратная реализация функций программного обеспечения. Такой подход является особенно актуальной для систем реального времени.

Особенность режима реального времени заключается в том, что на реакцию системы накладываются ограничения со стороны внешних факторов. Система должна отреагировать на события, происходящие на объекте управления, за определенный промежуток времени, величина которого определяется особенностями объекта управления. Выход за допустимые временные рамки приводит к снижению качества управления или потере управляемости объектом, что может повлечь нежелательные последствия, в том числе катастрофические.

Из широкого круга систем реального времени можно в первую очередь выделить весьма широкий класс системы управления объектами и различными технологическими процессами (системы числового программного управления, навигации и т.д.). Для таких систем характерным является необходимость решения траекторных задач, требующих вычисления различных функциональных зависимостей.

В связи с этим важной задачей является разработка аппаратных средств для вычисления различных функций, позволяющих обеспечить минимальное время вычисления функций.

Анализ методов, использующих таблицы функций

Высокую скорость воспроизведения функции обеспечивает табличный метод, основанный на хранении в памяти полной таблицы функции. В этом случае для получения значения функции достаточно одного обращения к памяти. Реализация такого похода требует наличия $K \cdot 2^n$ бит памяти, где K – число воспроизводимых функций, а n – разрядность операндов. При больших значениях K и n обеспечить такую емкость не всегда возможно.

С целью уменьшения аппаратурных затрат используют таблично-алгоритмические методы [1-3], которые сочетают обращение к таблицам и обработку полученной информации, например, с использованием полиномиальной аппроксимации. В связи с этим возникает необходимость разработки быстродействующих операционных устройств для вычисления многочленов.

Современные достижения в области интегральной технологии предоставляют средства, которые позволяют создавать сложные системы, в том числе, на одном кристалле (технология CSoC – Configurable System on Chip). Лидерами данного направления являются фирмы Triscend, Xilinx, Altera. В общем случае микросхемы содержат вычислительные ядра, память и программируемую логику FPGA (Field Programmable Gate Arrays), что

дает возможность эффективной реализации таблично-алгоритмических методов вычислений.

В статье [4] авторов данной работы проведен анализ эффективности различных таблично-алгоритмических методов вычисления многочленов и предложен метод, базирующийся на предварительной обработке коэффициентов. Показано, что метод обеспечивает вычисление многочлена

$$P(x) = \beta_0 + \beta_1 X + \beta_2 X^2 + ... + \beta_m X^m \qquad (1)$$

за время $T = t_C + 2t_П + t_Д$, где t_C – время сложения двух чисел, $t_П$ – время обращения к памяти, $t_Д$ – время сложения $(m+2)$ чисел с помощью дерева сумматоров (m – степень полинома). Показано, что предложенный метод позволяет уменьшить аппаратурные затраты по сравнению с известными таблично-алгоритмическими методами, причем, без снижения быстродействия.

В данной работе исследуется возможность ускорения вычислений с использованием предварительной обработки коэффициентов многочлена.

Обоснование метода

Представим многочлен (1) в виде

$$P_1(X) = \alpha_0 + (\alpha_1 + X)^2 + (\alpha_2 + X)^3 + \\ + ... + (\alpha_m + X)^{m-1} + F_1(X). \qquad (2)$$

Определим, при каких значениях $\alpha_0,...,\alpha_m, F_1(X)$ выполняется равенство

$$P_1(X) = P(X).$$

Разложив в (2) составляющие $(\alpha_j + X)^{j+1}$ для $j = \overline{1,m}$ в соответствии с формулой бинома Ньютона, получим

$$P_1(X) = \alpha_0 + \sum_{i=0}^{2} C_2^i \alpha_1^{2-i} X^i + \sum_{i=0}^{3} C_3^i \alpha_2^{3-i} X^i + ... + \\ + \sum_{i=0}^{j+1} C_{j+1}^i \alpha_j^{j+1-i} X^i + ... + \sum_{i=0}^{m+1} C_{m+1}^i \alpha_m^{m+1-i} X^i + F_1(X).$$

Выделим в полученном выражении составляющую $(X^2 + X^3 + ... + X^{m+1})$ и сгруппируем оставшиеся члены по степеням X:

$$P_1(X) = (\alpha_0 + \sum_{i=1}^{m} C_{i+1}^0 \alpha_i^{i+1}) + X \sum_{i=1}^{m} C_{i+1}^1 \alpha_i^i + \\ + X^2 \sum_{i=2}^{m} C_{i+1}^2 \alpha_i^{i-1} + ... + X^j \sum_{i=j}^{m} C_{i+1}^j \alpha_i^{i-j+1} + ... + \\ + C_{m+1}^m \alpha_m X^m + F_1(X) + (X^2 + X^3 + ... + X^{m+1}).$$

Приравняв полученное выражение исходному многочлену с учетом $C_k^0 = 1$, $C_{k+1}^k = k+1$, получим:

$$F_1(X) = -(X^2 + X^3 + ... + X^{m+1}),$$

$$\alpha_m = \beta_m / (m+1),$$

$$\alpha_j = (\beta_j - \sum_{i=j+1}^{m} C_{i+1}^j \alpha_j^{i-j+1}) / (j+1),$$

$$\alpha_1 = (\beta_1 - \sum_{i=2}^{m} C_{i+1}^1 \alpha_i^i) / 2,$$

$$\alpha_0 = \beta_0 - \sum_{i=1}^{m} \alpha_i^{i+1}.$$

Устройство, работающее в соответствии с выражением (2), может быть построено по схеме на рис. 1. Блоки $П_i$ ($i = \overline{1,m}$) предназначены для возведения чисел в степень $(i+1)$, а блок $П_{m+1}$ обеспечивает хранение таблицы функции $F_1(X)$. Сумматоры См осуществляют суммирование двух чисел, а суммирующий блок СБ – $(m+2)$ чисел.

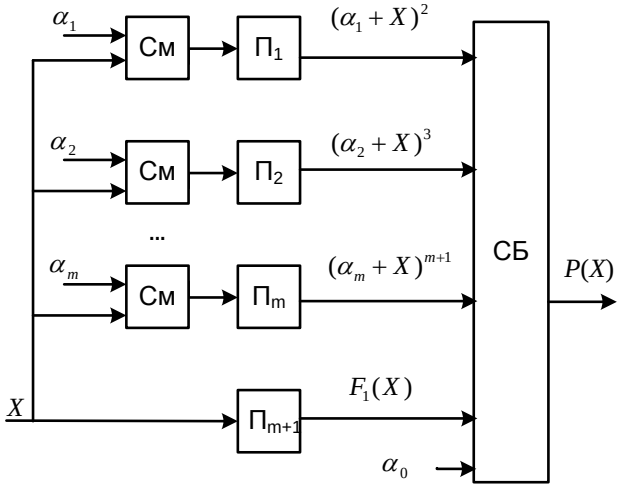

Рис. 1. Структура устройства

Общая емкость необходимой памяти определяется как

$$N = \sum_{i=1}^{m+1} 2^{n_{2i}} n_{1i}, \qquad (3)$$

где n_{1i} и n_{2i} – разрядность соответственно выходных и входных слов блоков Π_i.

Разрядность слов зависит от формата данных и требуемой точности вычислений. Рассмотрим эту зависимость.

Будем считать, что $n_{1i} = n'_{1i} + n''_{1i}$, $n_{2i} = n'_{2i} + n''_{2i}$, где n'_{1i} и n'_{2i} – разрядности целой части, n''_{1i} и n''_{2i} – разрядности дробной части соответственно выходных и входных слов.

Значения n'_{1i} и n'_{2i} для блоков Π_i $(i = \overline{1,m})$ определяются так:

$$\begin{aligned} n'_{1i} &=]\log_2 (\alpha_i + X)^{i+1}_{\max}[, \\ n'_{2i} &=]\log_2 (\alpha_i + X)_{\max}[. \end{aligned} \quad (4)$$

Здесь $]\cdot[$ – функция округления до ближайшего большего целого числа, если данное число не целое.

Для блока Π_{m+1} получим:

$$\begin{aligned} n'_{1i} &=]\log_2 \sum_{j=2}^{m+1} X^j[, \\ n'_{2i} &=]\log_2 X[. \end{aligned} \quad (5)$$

Очевидно, что величины n'_{1i} и n'_{2i} зависят от пределов изменения аргумента и коэффициентов α_i. Эти величины необходимо определить для всех многочленов, которые могут вычисляться на рассматриваемом устройстве. При расчете следует принять максимальные значения.

Разрядность дробной части выходных и входных слов блоков Π_i можно определить следующим образом:

$$\begin{aligned} n''_{1i} &=]\log_2 (1/\Delta_{1i})[, \\ n''_{2i} &=]\log_2 (1/\Delta_{2i})[, \end{aligned}$$

где Δ_{1i} и Δ_{2i} – погрешность представления соответственно выходных и входных слов.

Как следует из (2), значение $P_1(X)$ является результатом суммирования $(m+2)$ слагаемых. Так как погрешность Δ результата равна сумме погрешностей слагаемых, то приняв погрешности всех слагаемых равными Δ_1, найдем

$$\Delta_1 = \Delta/(m+2) = 2^{-n}/(m+2). \quad (6)$$

Тогда для блоков Π_i $(i = \overline{1, m+1})$ получим

$$n''_{1i} = n +]\log_2 (m+2)[. \quad (7)$$

Исходя из значения Δ_1, определим погрешность Δ_{2i} представления входных слов блоков $\Pi_1 - \Pi_m$.

Погрешность Δ_1 возведения в степень $(i+1)$ числа $Z_i = (\alpha_i + X)$, представленного с погрешностью Δ_{2i}, определяется как

$$\begin{aligned} \Delta_1 &= (Z_i + \Delta_{2i})^{i+1} - Z_i^{i+1} = C_{i+1}^1 Z_i^i \Delta_{2i} + \\ &+ C_{i+1}^2 Z_i^{i-1} \Delta_{2i}^2 + \ldots + C_{i+1}^i Z_i \Delta_{2i}^i + C_{i+1}^{i+1} \Delta_{2i}^{i+1}. \end{aligned}$$

Пренебрегая членами, содержащими второй и более высокие порядки Δ_{2i}, получим

$$\Delta_1 = C_{i+1}^1 Z_i^i \Delta_{2i} = (i+1) Z_i^i \Delta_{2i}.$$

Отсюда видно, что погрешность результата максимальна при максимальном значении $Z_i = (\alpha_i + X)\max$. С учетом (6) определим

$$\Delta_{2i} = 2^{-n}/((i+1)(m+2)(\alpha_i + X)^i_{\max}).$$

Исходя из этого, разрядность представления дробной части входного слова блока Π_i будет составлять

$$n''_{2i} = n +]\log_2 ((i+1)(m+2)(\alpha_i + X)^i_{\max})[. \quad (8)$$

Суммарную длину выходных и входных слов блоков Π_i $(i = \overline{1,m})$ определим с учетом (4), (7) и (8) так:

$$\begin{aligned} n_{1i} &= n +]\log_2 (\alpha_i + X)^{i+1}_{\max})[+ \\ &+]\log_2 (m+2))[, \\ n_{2i} &= n +]\log_2 (\alpha_i + X)_{\max})[+ \\ &+]\log_2 ((i+1)(m+2)(\alpha_i + X)^i_{\max})[. \end{aligned} \quad (9)$$

Разрядность выходных и входных слов блока Π_{m+1} получим из (5) и (7), с учетом того, что входное слово X блока Π_{m+1} представлено n разрядами после запятой. Тогда:

$$\begin{aligned} n_{1(m+1)} &= n +]\log_2 \sum_{j+2}^{m+1} X^j[+]\log_2 (m+2)[, \\ n_{2(m+1)} &= n +]\log_2 X[. \end{aligned} \quad (10)$$

Таким образом, суммарная емкость требуемой памяти для хранения таблиц определяется выражением (3) с учетом (9) и (10). Следует заметить, что аппаратурные затраты на реализацию данного метода и метода, рассмотренного в [4], имеют примерно одинаковый порядок.

Время вычислений в устройстве складывается из времени t_C сложения двух чисел, обращения к памяти $t_П$ и суммировании $(m+2)$-х чисел в СБ, то есть

$$T = t_C + t_П + t_Д .$$

Одним из наиболее быстродействующих блоков для сложения нескольких чисел является дерево сумматоров [5]. Если СБ построен в виде дерева сумматоров, то время вычисления многочлена составит

$$T = (\lceil \log_2 (m+2) \rceil + 1) t_C + t_П .$$

Таким образом, по сравнению с методом, рассмотренным в работе [4], предлагаемый метод позволяет сократить время вычислений на длительность одного цикла обращения к памяти.

Выводы

Предложенный метод вычисления многочленов позволяет по сравнению с наиболее эффективным из рассматриваемых известных таблично-алгоритмических методов ускорить воспроизведение функций. Уменьшение времени вычислений при прочих равных условиях позволяет расширить область применения систем управления в реальном времени за счет сокращения цикла обработки информации.

Для смены воспроизводимой функции достаточно один раз пересчитать коэффициенты полинома. Схема устройства при этом не претерпевает изменений.

Метод может быть эффективно использован в случае необходимости многократного вычисления значений функций при разных значениях аргумента. Такой режим вычислений является естественным, например, для систем управления, когда алгоритмы обработки информации в реальном времени, включающие вычисление полиномов, на протяжении кадра управления не изменяются.

Список литературы

1. Информационные системы: Табличная обработка информации / Под ред. Е.П.Балашова и В.Б.Смолова. – Л.: Энергоатомиздат, 1985. – 184 с.
2. Попов Б.А., Теслер Г.С. Вычисление функций на ЭВМ. Справочник. – К.: Наукова думка, 1984. – 600 с.
3. Кнут Д. Искусство программирования для ЭВМ: В 3-х т., т. 2. – М.: Мир, 1977. – 723 с.
4. Жабин В.И., Макаров В.В. Таблично-алгоритмический метод вычисления многочленов // Вісник НТУУ “КПІ”, Інформатика, управління та обчислювальна техніка. – № 46. – 2007. – С. 206-210.
5. Карцев М.А., Брик В.А. Вычислительные системы и синхронная арифметика. – М.: Сов. радио, 1981. – 360 с.