Раздел VI. Вычислительные комплексы нового поколения и нейрокомпьютеры

УДК 004.896

А.Н. Берёза, М.В. Ляшов

АППАРАТНАЯ РЕАЛИЗАЦИЯ НЕЛИНЕЙНЫХ МАТЕМАТИЧЕСКИХ ФУНКЦИЙ ДЛЯ НЕЙРОННЫХ СЕТЕЙ

Анализ существующих подходов аппаратной реализации математических функций. Развитие современной микроэлектроники позволяет на одном кристалле размещать устройства, которые раньше возможно было только реализовать программным способом. При этом аппаратная реализация различных математических функций по сравнению с программной имеет ряд преимуществ в части быстродействия и надежности.

Существует большое количество разнообразных методов вычисления элементарных функций. Из них в цифровой вычислительной технике нашли применение [1]: разложение в ряд Тейлора; аппроксимация с помощью различного вида полиномов; цепные дроби; рациональные приближения; табличные методы; итерационные методы.

При аппаратной реализации необходимо учитывать скорость вычисления функции (производительность), занимаемые аппаратные ресурсы, а также точность вычисления (сходимость).

Степенные полиномы (отрезок ряда Тейлора, полином Чебышева и т. д.) вычисляются чаще всего по схеме Горнера. Время, необходимое для вычисления полинома при разложении функции произвольного вида, приблизительно равно:

$$t_{\text{пол}} = m(t_{\text{умн}} + t_{сл}),$$

где m – степень полинома; $t_{\text{умн}}$ – время выполнения команды умножения; $t_{сл}$ – время выполнения команды сложения.

Преимуществом разложения в ряд Тейлора является возможность вычисления коэффициентов членов ряда непосредственно в процессе работы. В связи с этим отсутствует необходимость запоминания их в памяти машины, как это требуется при полиномиальной аппроксимации по Чебышеву. Однако такой способ определения коэффициентов требует значительного времени вычисления. Недостатком использования ряда Тейлора является также его медленная сходимость при вычислении функций $\ln x$, $arctg\ x$, $\arcsin x$, что приводит к большому времени вычисления и накоплению значительной инструментальной погрешности. Кроме того, при этом способе методическая погрешность монотонно возрастает с увеличением аргумента. Для уменьшения влияния этого вида погрешности аргумент предварительно сводят в более узкую область с помощью соответствующих преобразований.

Методическая погрешность полиномиальной аппроксимации знакопеременна и равномерно распределена по промежутку изменения аргумента. К недостаткам этого метода следует отнести большую загрузку ПЗУ, в которое должны быть записаны коэффициенты всех аппроксимирующих полиномов.

Преимуществом цепных дробей является малый объем требуемый ПЗУ, принципиально более широкая область сходимости, чем для ряда Тейлора, и универсальность применения программы (микропрограммы) вычисления цепной дроби для различных функций. Однако практически получается, что при увеличении аргумента функции резко возрастает необходимое число звеньев дроби, что заставляет приводить аргументы к интервалу, не более широкому, чем при разложении в ряд Тейлора. Недостатком метода является, во-первых, необходимость наличия в системе команд машины операции деления, во-вторых, большое время вычисления, которое даже для самого быстрого способа составляет:

$$t_{\text{ц.др}} = \nu(t_{\text{умн}} + t_{\text{дел}} + t_{сл}),$$

где ν – число звеньев дроби, зависящее от вида функции и требуемой точности.

При использовании рациональных приближений элементарные функции представляются в виде отношения двух полиномов с числом членов в каждом намного меньшим, чем для соответствующих разложений в ряд Тейлора. Однако при данном методе все коэффициенты полиномов должны быть предварительно занесены в память. Время вычисления элементарной функции при использовании рациональных приближений состоит из суммарного времени вычисления двух полиномов и выполнения операции деления.

Различные табличные методы основаны большей частью на методах криволинейной или кусочно-линейной аппроксимации. Достоинством их является отсутствие арифметических операций, необходимых для вычисления функции. Однако практическое их использование ограничивается требованием большого объема запоминающего устройства. Этот фактор ограничивает практическое применение рассматриваемого метода.

Метод CORDIC (в отечественной литературе также употребляются термины метод «цифра за цифрой» или алгоритм Волдера (Volder)). Название является сокращением (аббревиатурой) от английского словосочетания «COordinate Rotation DIgital Computer», что можно перевести как цифровое вычисление поворота координат. Идея метода заключается в сведении вычисления значений сложных (например, гиперболических) функций к набору простых шагов вида сложений, вычитаний и сдвига, т.е. переноса запятой. Такой подход особенно полезен при вычислении функций на устройствах с ограниченными вычислительными возможностями, такими как микроконтроллеры или программируемые логические матрицы (FPGA). Кроме того, поскольку шаги однотипны, то при аппаратной реализации алгоритм поддается развёртыванию в конвейер либо свертыванию в цикл.

Основное преимущество метода CORDIC состоит в том, что те же самые аппаратные средства могут быть использованы для нескольких функций, но при этом падает производительность.

Сущность итерационных методов состоит в построении последовательности $y_{i+1} = f(y_i)$, сходящейся к функции $y(x)$. Эффективность применения данных методов зависит, во-первых, от скорости сходимости (необходимого числа итераций, удовлетворяющего заданной точности) и, во-вторых, от набора операций, выполняющихся в каждом шаге. Необходимость в ряде случаев выполнения операций

деления и умножения в процессе выполнения итераций существенно увеличивает время вычисления.

Из всех рассмотренных выше методов при аппаратной реализации математических функций наиболее часто используется комбинация полиномов младших порядков совместно с табличными методами. Данный подход наиболее пригоден для ПЛИС, выполненных по технологии FPGA, поскольку микросхемы на кристалле содержат встроенные блоки памяти, а также умножители и сумматоры. Основное преимущество аппроксимации полиномами младшими порядками – те же самые аппаратные средства могут быть использованы для вычисления различных математических функций, при изменении только полиномиальных коэффициентов (т.е. содержание таблиц) [3].

При кусочно-линейной аппроксимации нелинейных функций возникают две проблемы. Первая – это выбор количества интервалов. Вторая – для нелинейных функций аппроксимация полиномами с постоянным шагом является не оптимальной, поскольку в зависимости от аргумента производная функции имеет различное значение.

Пример кусочно-линейной аппроксимации сигмоидальной функции активации. При создании нейронных сетей аппаратная реализация нелинейных функций активации является одной из наиболее сложных проблем. Наиболее распространенной функцией является нелинейная функция активации с насыщением, так называемая логистическая функция или сигмоид (функция S-образного вида):

$$f(x) = \frac{1}{1+e^{-ax}}.$$

При уменьшении a сигмоид становится более пологим, в пределе, при $a = 0$, вырождается в горизонтальную линию на уровне $0{,}5$, а при увеличении a сигмоид приближается к виду функции единичного скачка. Сигмоидальная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того, она обладает свойством усиливать слабые сигналы и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.

Пусть $I = [L,U]$ – интервал аппроксимации, где $L < U$ и $f : I \to R$ – функция которую аппроксимируем, где R – множество действительных чисел. Предположим, что $\hat{f} : I \to R$ – линейная функция $\hat{f}(x) = c_1 + c_2 x$. Относительная ошибка при этом равна

$$\varepsilon(x) = \frac{f(x) - \hat{f}(x)}{f(x)}.$$

Для нахождения коэффициентов c_1 и c_2, решая систему уравнений:

$$\begin{cases} c_1 + c_2 L = \dfrac{1}{1+e^{-aL}} \\ c_1 + c_2 U = \dfrac{1}{1+e^{-aU}} \end{cases},$$

получим:

$$c1 = \frac{\alpha}{\alpha+\beta},\ c2 = \frac{e^{-aL} - e^{-aU}}{\alpha+\beta},$$

где $\alpha = U - L + Ue^{-au} - Le^{-aL}$ и $\beta = Ue^{-aL} + Ue^{-aU-aL} - Le^{-aU-aL} - Le^{-aU}$.

Структурная схема устройства, реализующего кусочно-линейную аппроксимацию, показана на рис. 1.

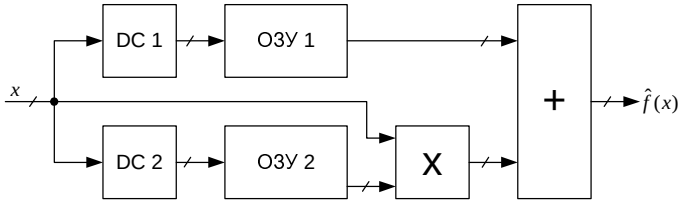

Рис. 1. Структурная схема устройства, реализующего линейную интерполяцию

Рассчитанные коэффициенты c_1 и c_2 хранятся в ОЗУ 1 и ОЗУ 2 соответственно. Применение ОЗУ позволяет динамически изменять функцию активации во время функционирования сети без выключения питания. Декодеры адреса DC 1 и DC 2 предназначены для декодирования аргумента x в соответствии с шагом аппроксимации.

На рис. 2 представлен алгоритм расчета оптимального шага аппроксимации.

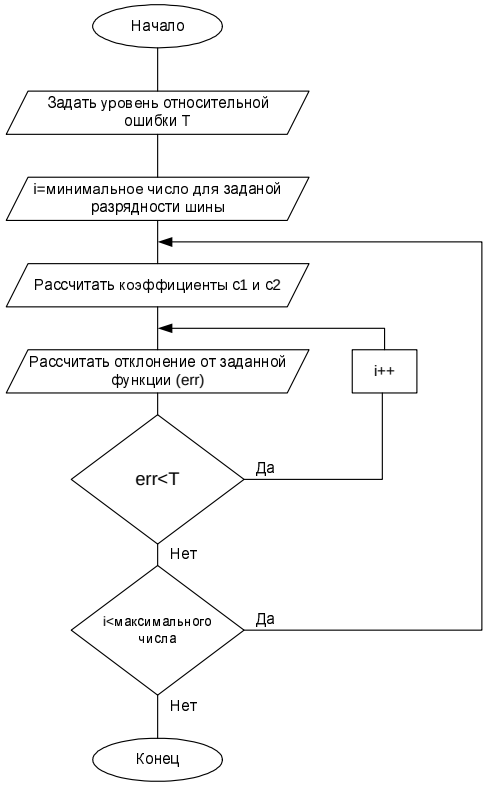

Рис. 2. Алгоритм расчет оптимального шага аппроксимации

На рис. 3 приведен пример аппаратной реализации 8-разрядного устройства линейной интерполяции, выполненный в системе автоматизированного проектирования программируемых логических интегральных схем Quartus II фирмы Altera.

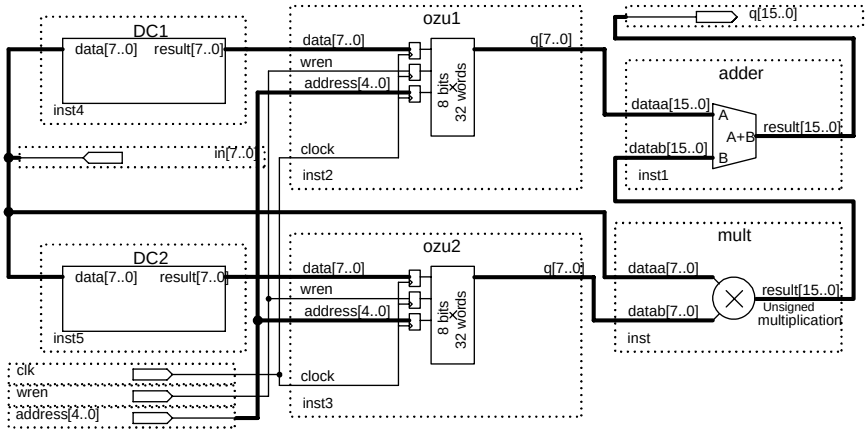

Рис. 3. Принципиальная схема устройства линейной интерполяции в файле верхнего уровня, созданная в пакете Quartus II

В результате моделирования работы устройства линейной интерполяции в системе автоматического проектирования Quartus II тактовая частота составляет 240 МГц. Один модуль сумматора занимает в микросхеме семейства Strarix EP1S10F484C5 фирмы Altera менее 2%.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Байков В.Д. Сомолов В.Б.* Аппаратурная реализация элементарных функций в ЦВМ. – Л.: Изд-во Ленингр. ун-та, 1975. – 96 с.
2. *Бандман О.А.* Специализированные процессоры для высокопроизводительной обработки данных. – Новосибирск: Наука, 1988. – 204 с.
3. *Шевкопляс Б.В.* Микропроцессорные структуры. Инженерные решения. Справочник / 2-е изд. перераб. и доп. – М.: Радио и связь, 1990. – 512 с.
4. *Партала О.Н.* Цифровая электроника / Издание 2-е, дополненное – СПб: Наука и Техника, 2001. – 224 с.

УДК 681.518

В.А. Литвиненко, С.А. Ховансков, О.Р. Норкин

АЛГОРИТМ УСКОРЕНИЯ ВЫПОЛНЕНИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ В КОМПЬЮТЕРНОЙ СЕТИ*

Существует класс задач требующих большого объема вычислений и имеющих жесткое ограничение по времени выполнения. Традиционно такие задачи решаются путем организации распределенных вычислений на нескольких центрах обработки данных вычислительной системы [1, 2].

В качестве вычислительной системы используется либо мультипроцессорная вычислительная машина, либо многомашинная вычислительная система, либо обычная компьютерная сеть, обладающая достаточным или избыточным количеством центров обработки данных (локальная, глобальная сети). При этом вычислительная

* Работа выполнена при поддержке РФФИ (грант № 07-01-00174).