ОДНОРОДНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
(Вычислительные системы)
1981 год Выпуск 90

УДК 681.324

ОРГАНИЗАЦИЯ ТАБЛИЧНЫХ ВЫЧИСЛЕНИЙ НА ОСНОВЕ СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕД

П.А. Анишев

Введение

В настоящее время в связи с развитием микроэлектроники становится реальным производство программируемых однородных структур (вычислительных сред) большого объема. В связи с этим все большее развитие получают методы вычисления с использованием таблиц, а также методы, использующие логарифмы [1-6].

Существует целый ряд задач (управление движением роботов-манипуляторов, цифровая обработка сигналов и др.),где нужны быстрые вычисления значений элементарных функций при малой разрядности (10-12 двоичных разрядов). Использование специализированных сред для таких вычислений должно основываться на правильном выборе метода хранения и/или способа вычислений,что, в свою очередь, зависит от таких факторов,как требуемая точность представления чисел, время получения результатов, объем оборудования и т.д.

В статье предлагается способ кодирования (аналогичный рассмотренному в [7]) значений одноместной монотонной функции,ориентированный на реализацию в специализированной вычислительной среде, и рассматривается применение этого способа для представления некоторых элементарных функций. Производятся сравнительные оценки памяти и времени для предлагаемого и других способов организации вычислений.

§1. Метод приращений

Если непрерывная функция $f(x)$ имеет в интервале (a,b) производную $f'(x)$ такую, что $0 \le f'(x) \le 1$ для всех $x \in (a,b)$ и $f(a) = 0$, то $f(x)$ может быть представлена (с заданной точностью Δ) вектором из нулей и единиц длины $\left]\frac{|a-b|}{\Delta}\right[$ в виде ступенчатой функции $\bar{f}(x)$. Значение $f(x)$ выбирается так, чтобы отклонение $|\bar{f}(x) - f(x)|$ было минимальным (см. рис.I). Высота ступеньки равна заданной точности Δ. Интервал (a,b) разбит на дискретные значения $a = x_1, x_2, \dots, x_n = b$. Вектор, кодирующий функцию $\bar{f}$, формируется следующим образом: i-я координата равна 1, если в точке x_i ступенчатая функция $\bar{f}$ имеет разрыв, и 0 – в противном случае. Значение функции $\bar{f}$ в точке x_i равно числу единиц в отрезке вектора от 1 до i, умноженному на Δ. В качестве значения обратной функции f^{-1} в точке y_k можно принять номер k-й единицы (умноженный на Δ) в векторе, кодирующем $\bar{f}$. Таким образом, один и тот же вектор может быть использован для представления пары взаимно-обратных функций. Рассмотрим пример на рис.I. Вектор, кодирующий функцию $f(x)$, имеет вид: II0I00I0I00, значение $f(x_7) \approx \bar{f}(x_7) = 4\Delta$; $f(x_5) \approx f(x_6) \approx \bar{f}(x_5) = 3\Delta$; $f^{-1}(y_4) = 7\Delta$; $f^{-1}(y_5) = 9\Delta$; высота ступеньки равна Δ.

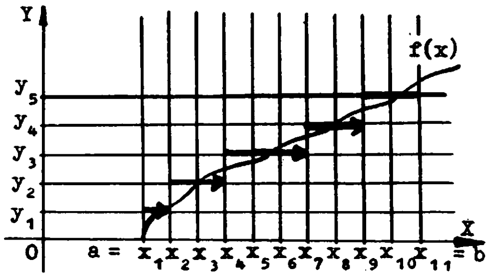

Рис. I

Для вычисления значений наиболее часто используемых элементарных функций достаточно иметь таблицы для двух пар взаимно-обратных функций ($\sin\frac{\pi}{2}x$ и $\frac{2}{\pi}\arcsin x$; $\log_2(1+x)$ и 2^x-1). Областью определения будем считать полуинтервал $[0,1)$, т.е. число с фиксированной запятой от 0 до $1-2^{-n}$. Так как производные этих функций не во всей области определения меньше единицы, то для представления этих функций методом приращений можно использовать следующий прием. Если производная функции $f(x)$ в интервале (a,b) удовлетворяет условию $1 \le f'(x) \le 2$, то вместо функции $f(x)$ можно методом приращений представить функцию $f(x)-x$, т.е. фактически в векторе закодировано отклонение графика функции от диагонали квадрата (см. рис.2).

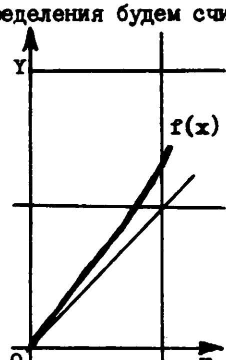

Рис. 2

При представлении чисел с фиксированной запятой обычно табулируются функции $2^{x}-1$, $\log_2(1+x)$, $\sin\frac{\pi}{2}x$, для которых в области $0 \le x < 1$ максимальное значение производной не превосходит 2:

$$\max_{x\in[0,1)} (2^{x}-1)' = 2\ln 2 \approx 1{,}386 < 2;$$

$$\max_{x\in[0,1)} (\log_2(1+x))' \approx 1{,}442 < 2;$$

$$\max_{x\in[0,1)} (\sin\frac{\pi}{2}x)' \approx 1{,}572 < 2.$$

Область определения [0,I) для каждой из этих функций может быть разбита на две части: в одной из них значение производной меньше I, в другой - меньше 2.

Итак, используя метод приращений, каждую из рассматриваемых функций можно представить двоичным вектором длиной 2^n бит. При этом точность представления значений аргумента и функции будет 2^{-n}. Задача подсчета в заданном двоичном векторе для каждого номера разряда i числа единиц в отрезке вектора от 1 до i может быть решена различными путями. Алгоритмы для решения этой задачи в обычных ЭВМ рассмотрены в [8], а в работах [9,I0] предлагаются устройства, которые могут быть использованы для этой цели. В устройстве, описанном в [9], для двоичного вектора производится как подсчет числа единиц, так и вычисление номера k-й единицы, что позволяет вычислить значения прямой и обратной функций, закодированных этим вектором. Для обработки вектора в N бит такое устройство должно содержать N^2 ячеек. Устройство, описанное в работе [I0], представляет собой однородный параллельный сумматор. Для обработки вектора длиной $N=2^n$ бит сумматор должен содержать $M = \sum_{i=1}^{n} i\cdot 2^{i-1}$ ячеек, каждая из которых является одноразрядным сумматором с памятью. Операцию подсчета единиц на параллельном сумматоре можно осуществить за $2\sqrt{N}$ тактов при размещении ячеек на плоскости, и за $3\sqrt[3]{N}$ - в пространстве.

§2. Сравнительные оценки памяти и времени

Проведем сравнение четырех методов вычисления значений элементарных функций: метода приращений; линейной интерполяции; прямого хранения; непосредственного вычисления путем разложения в степенные ряды.

Введем обозначения:

n - количество двоичных разрядов в представлении числа;

$\Delta = 2^{-n}$ - точность представления чисел;

m - число значений функции, хранимых в таблице.

В дальнейшем будем считать, что одно обращение к таблице, сдвиг на один разряд и такт работы суматора приблизительно одинаковы и равны t . Через $\|A\|$ обозначим сложность вычисления величины A в единицах t .

Сложение двух n-разрядных чисел может быть выполнено за время распространения переноса, в среднем равное ($t \log_2 n$(см. [II]). Умножение двух n-разрядных чисел с помощью сложений и сдвигов может быть выполнено в среднем за $\frac{n}{2}\log_2 n+n$ ($\frac{n}{2}$ сложений и n сдвигов) тактов.

I. Оценка метода приращений. При точности 2^{-n} объем таблицы равен 2^n бит, время получения значения функции F равно $\|F\| = 3\, 2^{\frac{n}{3}} t$ (с использованием параллельного суматора [IO]).

2. Оценка метода линейной интерполяции. Для оценки числа диапазонов разбиения области определения функции при точности 2^{-n} воспользуемся теоремой из [12].

ТЕОРЕМА. Если функция $f(x)$ имеет в некотором интервале (a,b) непрерывные производные до $(k+1)$-го порядка и если $P_k(x) = f(x)$ для $x = x_0, x_1, \ldots, x_k \in (a,b)$, то

$$\Delta = f(x) - P_k(x) = \frac{f^{(k+1)}(s)(x-x_0)(x-x_1)\ldots(x-x_k)}{(k+1)!},$$

где $P_k(x)$ - полином порядка k, а

$$\min_i\{x_i\} \le s \le \max_i\{x_i\}.$$

Для функций $\sin x, \arcsin x, \log_2 x, 2^x$ число диапазонов можно взять равным $\sqrt{2^n}$ при точности 2^{-n}. Это удобно и с той точки зрения, что за табличные значения можно принять (при разбиении на равные диапазоны) старшие $\frac{n}{2}$ разрядов числа X

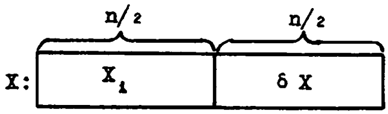

Рис. 3

(см.рис.3). Итак, объем памяти для таблицы при линейной интерполяции составляет $n\sqrt{2^n}$ бит. Вычисление значения функции F в промежуточной точке X можно осуществить (см. рис.4) по формуле из [I]:

$$F(X) = F(X_i) + \frac{[F(X_{i+1}) - F(X_i)] \cdot \delta X}{X_{i+1} - X_i}, \qquad (1)$$

где X_i – число, образованное старшими $\frac{n}{2}$ разрядами числа X, младшие $\frac{n}{2}$ разрядов – нули; δX – число, образованное младшими $\frac{n}{2}$ разрядами числа X, старшие $\frac{n}{2}$ разрядов – нули. Деление на $X_{i+1} - X_i$ может быть заменено сдвигом на $\frac{n}{2}$ разрядов, X_{i+1} получается из X_i прибавлением единицы в $\frac{n}{2}$-й разряд.

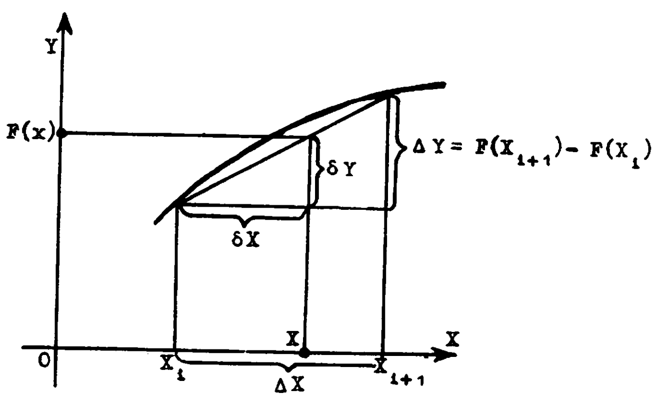

Рис. 4

Таким образом, число тактов для получения значения функции по формуле (I) равно $2 + 3\log_2 n + \frac{n}{2}\log_2 n + n + \frac{n}{2}$ (два обращения к таблице, три сложения, умножение и сдвиг на $\frac{n}{2}$ разрядов).

3. При прямом хранении требуется $n2^n$ бит памяти и одно обращение к таблице.

4. Прямое вычисление при разложении элементарных функций в степенные ряды требует для точности 2^{-12} – три, а для точности 2^{-16} – пять членов ряда [I3], что составит для точности 2^{-12} около I00 тактов (три умножения и три сложения): $3(\frac{12}{2}\log_2 12 + 12) + 3 \cdot 4 \approx 100$, а для точности 2^{-16} – 260 тактов (пять умножений и пять сложений): $5(\frac{16}{2}\log_2 16 + 16) + 5 \cdot 4 = 260$.

Объемом памяти, отводимой под коэффициенты ряда, можно пренебречь.

Таблица I

Методы	n = 12		n = 16		Наличие памяти микропрограмм
	память (биты)	время (такты)	память (биты)	время (такты)	
Метод приращений	2^{12}	48	2^{16}	I20	-
Линейная интерполяция	$12 \cdot 2^6$	50	2^{12}	70	есть
Прямое хранение	$12 \cdot 2^{12}$	I	2^{20}	I	-
Прямое вычисление	-	I00	-	260	есть

Результаты проведенного сравнения для точности 2^{-12} и 2^{-16} отражены в табл.I. Анализируя ее, можно сделать вывод о том, что применение метода приращений оправдано при малой (I0-I2) разрядности: время получения результата совпадает с временем при линейной интерполяции, а возрастание объема памяти компенсируется упрощением операционной части, так как отпадает необходимость в операции умножения и в памяти микропрограмм для этой операции.

Литература

1. AUS H.M., KORN G.A. Table-look up/interpolation function generation for fixed point digital computations. - IEEE Trans.Comput.,1969,v.C-18, N 8,p.745-749.

2. BRUBAKER T.A., BECKER J.C. Multiplication using logarithms implemented with read-only memory. - IEEE Trans.Comput.,1975, v.C-24,N 8,p.761-765.

3. SWARTZLANDER E.E. et al. The sign/logarithm number system.-IEEE Trans.Comput.,1975,v.C-24, N 12,p.1238-1242.

4. JAYASHREE T., BASU D. On binary multiplication using the quarter squre algorithm.- IEEE Trans. Comput., 1976, v.C-25,N 9, p.957-960.

5. MITCHELL I.N. Computer multiplication and division using binary logarithms. - IRE Trans.Elect.Comp., 1962, v.EC-11, N 8, p.512-517.

6. ГОЛЬДЕНБЕРГ Л.М., БУТЫЛЬСКИЙ Ю.Т., ПОЛЯК М.Н. Цифровые устройства на интегральных схемах в технике связи. -М.: Связь, 1979. - 232 с.

7. КАЛЯЕВ А.В. Введение в теорию цифровых интеграторов.-Киев: Наукова думка, 1964. - 291 с.

8. РЕЙНГОЛЬД Э., НИВЕРГЕЛЬТ Ю., ДЕО Н. Комбинаторные алгоритмы. Теория и практика. - М.: Мир, 1980. - 480 с.

9. ФЕТ Я.И. Преобразование структур данных в специализированных однородных процессорах. -Управляющие системы и машины, 1980, № 4, с.32-38.

10. КОРНЕВ Ю.Н., ПИСКУНОВ С.В., СЕРГЕЕВ С.Н. Двоичный накопительный сумматор. Авт. свид. № 436350. - Бюл. изоб., 1974, № 26.

11. БЕРКС А., ГОЛДСТЕЙН Г., НЕЙМАН Дж. Предварительное рассмотрение логической конструкции электронного вычислительного устройства. - Кибернетический сборник, 1964, № 9, с. 22-24.

12. ЛАНЦОШ К. Практические методы прикладного анализа. - М.: ИЛ, 1961. - 500 с.

13. КОРН Г., КОРН Т. Справочник по математике. - М.: Наука, 1974. - 832 с.

Поступила в ред.-изд.отд.
25 марта 1981 года