УДК 004.31

© 2008 р. В.В. Макаров, В.В. Жабина

Национальный технический университет Украины
"Киевский политехнический институт", г. Киев

СОВМЕЩЕНИЕ ВВОДА И ОБРАБОТКИ ОПЕРАНДОВ ПРИ ВЫЧИСЛЕНИИ НЕКОТОРЫХ ФУНКЦИЙ

Рассматривается метод вычисления функций синуса и косинуса в избыточной системе счисления при поразрядном вводе операндов, начиная со старших разрядов. Показана возможность совмещения во времени процессов ввода и обработки данных. Приведен алгоритм и пример вычислений. Исследована погрешность получения результата.

The method of sine and cosine function value calculation with digit-by-digit input of operands beginning with the top digits in redundant numerical notation is considered. The opportunity of overlapping in time of processes of data input and processing is shown. The algorithm and example of calculations is given. The measure of inaccuracy of result calculation is investigated.

Введение

При проектировании цифровых вычислительных устройств, работающих в системах управления процессами в реальном масштабе времени, возникает задача уменьшения продолжительности обработки операндов, формирующихся поразрядно вне устройства. Режим работы операционных блоков в указанном случае является неавтономным, так как период поступления очередных разрядов операндов определяется быстродействием источников информации, то есть внешними по отношению к операционным устройствам (ОУ) факторами. Для уменьшения времени получения результата целесообразно использовать методы, которые позволяют совмещать во времени процессы ввода операндов и их обработки. Для выполнения ряда операций такие методы известны [1]. Ниже показана возможность вычисления в указанном режиме тригонометрических функций синуса и косинуса. При этом рассматривается возможность обработки аргументов как в избыточном коде, который используется в квазипараллельных ОУ [1], так и в каноническом, который можно рассматривать как частный случай избыточного представления чисел. Предлагаемый подход ориентирован на аппаратную реализацию ОУ, в том числе на использования ПЛИС.

Методы вычисления функций

Для вычисления функций используются аналитические методы различным типом аппроксимации функций [2]. Такие методы, как правило, реализуются на программном уровне и требуют больших временных затрат. Ускорение вычислений могут обеспечить табличные методы, основанные на хранении значений функций в памяти [3]. Однако при большой разрядности аргументов аппаратурные затраты быстро возрастают. Объем памяти для прямой табличной реализации должен составлять $2^n n$ бит, где n – разрядность аргумента. При больших n использование прямой таблицы, как правило, не представляется возможным без каскадирования устройств памяти, что усложняет реализацию ОУ и снижает быстродействие.

Альтернативным подходом являются таблично-алгоритмические методы вычисления функций, которые используют таблицы коэффициентов значительно меньшей емкости, чем табличные методы, и требуют выполнения меньшего числа операций по сравнению с алгоритмическими методами. Наиболее часто для аппаратной реализации используют методы «цифра за цифрой», например, метод Волдера [4, 5].

Вычисление функций $Y = \sin\Phi$ и $X = \cos\Phi$ методом Волдера в параллельных ОУ, когда код аргумента представлен перед началом операции всеми разрядами, выполняется в соответствии с итерационными формулами [5]:

$$\Phi_{i+1} = \Phi_i - \varepsilon_i \operatorname{arctg} 2^{-i};$$

$$sign\,\varepsilon_i = sign\,\Phi_i;$$

$$Y_{i+1} = Y_i - \varepsilon_i 2^{-i} X_i;$$

$$X_{i+1} = X_i + \varepsilon_i 2^{-i} Y_i.$$

Здесь $i = 0...n-1,\ Y_0 = 0,\ X_0 = 1/K$, $\Phi_0 = \Phi$, $\varepsilon_i \in \{-1,1\}$, K – коэффициент деформации вектора, равный $\Pi_0^{n-1}\sqrt{1+2^{-2i}}$, $Y_i = -\sin\Phi_i$ и $X_i = \cos\Phi_i$ – промежуточные результаты вычислений.

Можно выделить два этапа выполнения операции: преобразование аргумента (две первые формулы) и непосредственно нахождения значений функций.

Совмещать во времени процессы поразрядного ввода операндов и их обработки в данном случае не представляется возможным. К началу вычислений должен быть накоплен полный код аргумента. Таким образом, при поразрядном поступлении операнда время Т, необходимое для получения результата, определяется выражением

$$T = T_1 + T_2, \qquad (1)$$

где T_1 – время поразрядного ввода операндов, а T_2 – время собственно вычисления функции.

Таким образом, возникает необходимость разработки методов, которые позволяют ускорить время получения результата за счет устранения составляющей T_2 в выражении (1), в том числе, при использовании избыточных систем счисления.

Теоретическое обоснование предлагаемого метода

Пусть операнд представлен в двоичной позиционной избыточной системе счисления с цифрами $\{-1,0,1\}$ и изменяется в пределах $0 \le \Phi \le \pi/2$. Рассмотрим модификацию метода Волдера для случая, когда операнд поступает в ОУ последовательным кодом, начиная со старших разрядов.

Поскольку при этом в i-м цикле учитываются только поступившие старшие разряды операнда, то одной итерации в цикле может оказаться недостаточно, так как значение операнда может измениться при поступлении следующих разрядов. «Неправильная» итерация i-го цикла должна быть исправлена в последующих циклах.

Учитывая пределы изменения значений аргументов, для функций $Y = \sin\Phi$ и $X = \cos\Phi$ условие исправляемости итераций можно получить в виде

$$\left|\varphi_{i+1}2^{-i-1}\right| + \operatorname{arctg} 2^{-i} \le k\sum\nolimits_{j=i+1}^{n} arctg 2^{-k},$$

(2) где φ_{i+1} – очередная цифра операнда, $\operatorname{arctg} 2^{-i}$ – максимальный остаток от преобразования операнда в i-м цикле, k – максимальное количество итераций, которое необходимо выполнить в каждом цикле, чтобы скомпенсировать остаток предыдущего цикла и цифру операнда, поступившую в текущем цикле.

Можно показать, что неравенство (2) выполняется при $k = 2$, то есть в каждом цикле достаточно выполнить две итерации. Заметим, что в связи с использованием двойных итераций изменится коэффициент деформации вектора. В данном случае он будет иметь вид $K^{'} = K^2 = \Pi_0^n(1+2^{-2i})$.

При вычислении X_{i+1}, Y_{i+1} разряды результата можно формировать последовательно для их обработки в других ОУ, работающих в аналогичном режиме. Чтобы разряды результата x_{i+1}, y_{i+1}, имеющие вес 2^{-i-1}, не превышали максимальное допустимое значение, вывод результата необходимо производить с задержкой на p шагов, которая (например, для переменной X) определяется из условия

$$(x_{(i+1)\max} + 1)2^{-i-1+p} > (X_{i+1} - X_i) + X_i^{'}, \qquad (2)$$

где $X_i^{'}$ – код, содержащий только $(n-i-1+p)$ младших разрядов X_i, $x_{(i+1)\max}$ - максимальное значение разряда результата, в данном случае $x_{(i+1)\max} = 1$.

С учетом максимального значения $X_i^{'} = 2^{-i-1+p} - 2^{-n}$ неравенство (3) принимает вид

$$2^{-i-1+p} > (X_{i+1} - X_i) - 2^{-n}. \qquad (2)$$

При вычислении X_{i+1} разность

$(X_{i+1} - X_i)$ максимальна, если в i-м цикле дважды выполняется сложение или вычитание. В этом случае $(X_{i+1} - X_i)$ = $(2^{-i+1} Y_i - 2^{-2i} X_i)$. При максимальном значении $Y_i = 1$ и минимальном значении $X_i = 0$ эта разность составит $(X_{i+1} - X_i)$ = 2^{-i+1}. В этом случае неравенство (4) выполняется при $p = 2$.

Алгоритм и пример вычислений

Рассмотрим пример вычисления функций $Y = \sin\Phi,\ X = \cos\Phi$.

Устройство для вычисления функций должно содержать сумматоры и регистры для формирования и хранения операнда Φ и результатов Y и X. Кроме того, необходим блок памяти, в котором записаны константы вида $\operatorname{arctg} 2^{-i}$, а также средства коммутации.

Перед началом вычислений $X_0 = 1/K'$ = $(0.36875624)_{10}$ = $(0.010111101)_2$, $Y_0 = 0$, $\Phi_0 = \varphi_0 \cdot 2^0$. Каждый i-й цикл состоит из следующих тактов:

1 такт. На вход устройства поступает $(i+1)$-й разряд операнда $\varphi_{i+1} \in \{-1, 0, 1\}$ и формируется частичное значение операнда $\Phi'_{i+1} = \Phi_i + \varphi_{i+1} \cdot 2^{-i-1}$.

2 такт. Выполняются вычисления $\Phi''_{i+1}, X'_{i+1}, Y'_{i+1}$ и $sign\,\varepsilon_i = sign\,\Phi'_{i+1}$.

3 такт. Выполняются вычисления $\Phi_{i+1}, X_{i+1}, Y_{i+1}$ и $sign\,\varepsilon_i = sign\,\Phi''_{i+1}$.

Пусть, например, необходимо вычислить значение функций $Y = \sin\Phi,\ X = \cos\Phi$ при разрядности $n = 8$ и значении операнда Φ = $(0.7073)_{10}$ = $(0.110\bar{1}011\bar{1})_2$. Вычисления иллюстрируются табл. 1, в которой i - номер цикла, а j - номер такта.

В результате вычислений формируются представленные параллельным кодом значения $X = \cos\Phi =$ $(0.11000001)_2$ = $(0.7539)_{10}$ и $Y = -\sin\Phi$ = $(1.01011000)_2$ = $-(0.6562)_{10}$.

Погрешность вычислений

Погрешности первого этапа вычислений (преобразование аргумента), включают методическую погрешность конечного числа итераций и инструментальную погрешность суммирования округленных констант. Методическая погрешность не изменяется по сравнению с параллельными ОУ и составляет $\sigma_м = (2/\sqrt{3}) \cdot 2^{-n}$. Инструментальная погрешность с учетом двойных итераций составит $\sigma_и = (\sqrt{2n}/2\sqrt{3}) \cdot 2^{-n}$, а общая среднеквадратическая погрешность первого этапа равна

$$\sigma_1 = \sqrt{\sigma_м^2 + \sigma_и^2} = \sqrt{(n+8)/6} \cdot 2^{-n}. \quad (5)$$

Погрешности второго этапа вычислений (непосредственное вычисление функций) включают в себя погрешность, трансформированную из первого этапа и инструментальную погрешность вследствие сдвигов на i разрядов вправо. Трансформированная погрешность с учетом того, что величина угла может принимать значения в интервале от 0 до $\pi/2$, с учетом (5) составит σ_m = $(\sqrt{2}/2) \cdot \sqrt{(n+8)/6} \cdot 2^{-n}$. Инструментальная погрешность от сдвигов с учетом двойных итераций равна σ_c = $\sqrt{2(n-2)/3} \cdot 2^{-n}$, а общая погрешность вычисления функций составляет $\sigma = \sqrt{\sigma_m^2 + \sigma_c^2}$ = $\sqrt{3n/4 - 2/3} \cdot 2^{-n}$. В рассмотренном примере (табл. 1) погрешность при вычислении обеих функций не превышает расчетной.

Выводы

Предложенная модификация метода Волдера позволяет совмещать процессы ввода и обработки операндов, то есть позволяет устранить составляющую T_2 в выражении (1), что создает предпосылки к ускорению получения результата при поразрядном поступлении операндов.

Благодаря последовательному вводу информации сокращается необходимое число внешних выводов ОУ, что дает дополнительные преимущества по сравнению с ОУ параллельного типа. В частности, при использовании ПЛИС благодаря этому сокращается необходимое число ячеек ввода-вывода. В свою очередь, увеличение числа свободных выводов ПЛИС позволяет использовать их для других устройств в составе этой же микросхемы, то есть повышает возможности использования функционального ресурса микросхемы.

Таблица 1. Значения промежуточных результатов при вычислении функций

i	j	φ_{i+1}	arctg 2^{-i}	Φ_i	ε_i	X	Y
0	1	1		00.000000000			
				+00.100000000			
	2			00.100000000	1	0.010111101	0.000000000
			0.110010010	-00.110010010		+0.000000000	-0.010111101
	3			11.101101110	-1	0.010111101	1.101000011
				+00.110010010		-1.101000011	+0.010111101
				00.100000000		0.101111010	0.000000000
1	1	1		00.100000000			
				+00.010000000			
	2			00.110000000	1	0.101111010	0.000000000
			0.011101101	-00.011101101		+0.000000000	-0.010111101
	3			00.010010011	1	0.101111010	1.101000011
				-00.011101101		+1.110100001	-0.010111101
				11.110100110		0.100011011	1.010000110
2	1	0		11.110100110	-1	0.100011011	1.010000110
			0.001111101	+00.001111101		-1.110100001	+0.001000110
	2			00.000100011	1	0.101111010	1.011001100
				-00.001111101		+1.110110011	-0.001011110
	3			11.110100110		0.100101101	1.001101110
3	1	-1		11.110100110			
				-00.000100000			
	2			11.110000110	-1	0.100101101	1.001101110
			0.000111111	+00.000111111		-1.111001101	+0.000100101
	3			11.111000101	-1	0.101011111	1.010010011
				+00.000111111		-1.111010010	+0.000101011
				00.000000100		0.110001101	1.010111110
4	1	0		00.000000100	1	0.110001101	1.010111110
			0.000100000	-00.000100000		+1.111101011	-0.000011000
	2			11.111100100	-1	0.101111000	1.010100110
				+00.000100000		-1.111101010	+0.000010111
	3			00.000000100		0.110001110	1.010111101
5	1	1		00.000000100			
				+00.000001000			
	2			00.000001100	1	0.110001110	1.010111101
			0.000010000	-00.000010000		+1.111110101	-0.000001100
	3			11.111111100	-1	0.110000011	1.010110001
				+00.000010000		-1.111110101	+0.000001100
				00.000001100		0.110001110	1.010111101
6	1	1		00.000001100			
				+00.000000100			
	2			00.000010000	1	0.110001110	1.010111101
			0.000001000	-00.000001000		+1.111111010	-0.000000110
	3			00.000001000	1	0.110001000	1.010110111
				-00.000001000		+1.111111010	-0.000000110
				00.000000000		0.110000010	1.010110001
7	1	-1		00.000000000			
				-00.000000010			
	2			11.111111110	-1	0.110000010	1.010110001
			0.000000100	+00.000000100		-1.111111101	+0.000000011
	3			00.000000010	1	0.110000101	1.010110100
				-00.000000100		+1.111111101	-0.000000011
				11.111111110		0.110000010	1.010110001

СПИСОК ЛИТЕРАТУРЫ

1. *Жабин В.И., Корнейчук В.И., Тарасенко В.П.* Некоторые машинные методы вычисления рациональных функций многих аргументов // Автоматика и телемеханика. – 1977. – **12**. – С. 145-154.
2. *Попов Б.А., Теслер Г.С.* Вычисление функций на ЭВМ. – К.: Наукова думка, 1984. – 599 с.
3. *Корнейчук В.И. Запоминающие устройства ЦВМ. – К.: Техніка, 1976. – 168 с.*
4. *Volder J.E.* The CORDIC trigonometric computing technigue // IRE Trans. on Electr. Comp. – 1959. – Vol. **8**, No **3**. – P. 330-334.
5. *Байков В.Д., Смолов В.Б.* Специализированные процессоры: итерационные алгоритмы и структуры. – М.: Радио и связь, 1985. – 288 с.

DATA INPUT AND PROCESSING OVERLAPPING FOR SOME FUNCTION VALUE CALCULATION

СУМІЩЕННЯ ВВЕДЕННЯ І ОБРОБКИ ОПЕРАНДІВ ПРИ ОБЧИСЛЕННІ ДЕЯКИХ ФУНКЦІЙ

Розглядається метод обчислення функцій синуса і косинуса в надлишковій системі числення при порозрядному введенні операндів, починаючи зі старших розрядів. Показано можливість суміщення в часі процесів введення і обробки даних. Наведено алгоритм і приклад обчислень. Досліджено погрішність одержання результату.

Авторы

Макаров В.В. – к.т.н., с.н.с.
Жабина В.В. – аспирант
Кафедра специализированных компьютерных систем НТУУ „КПИ”